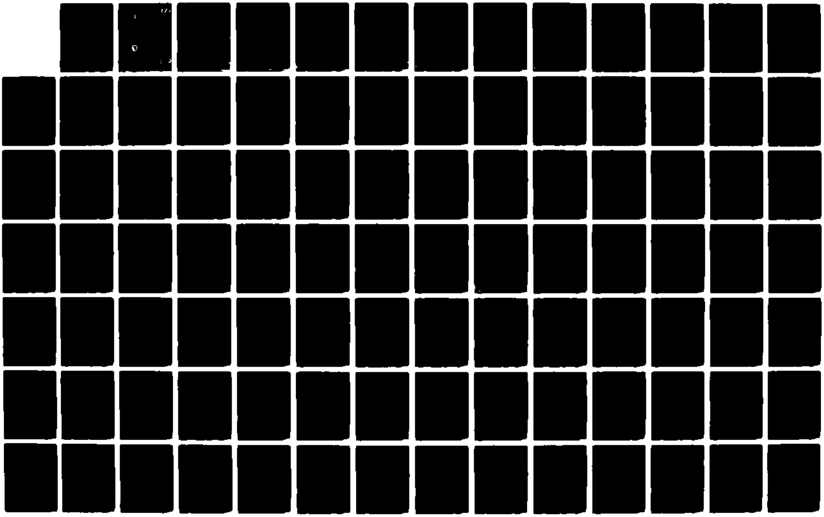
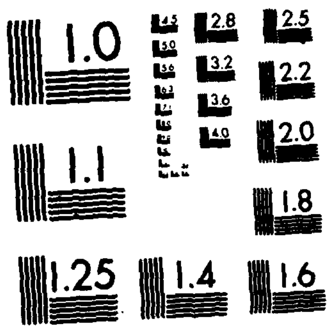
AD-A136 802 LOGAM (LOGISTIC ANALYSIS MODEL) VOLUME 3
TECHNICAL/PROGRAMMER MANUAL(U) COCKERHAM (JOHN M) AND
ASSOCIATES INC HUNTSVILLE AL AUG 82 1/3

UNCLASSIFIED USAMICOM-DA-SR-D-82-2-VOL-3 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A136802

Special Report D-82-2

LOGAM
TECHNICAL/PROGRAMMER MANUAL
VOLUME III

Systems Analysis Division
Systems Analysis and Evaluation Office
US Army Missile Command
Redstone Arsenal, Alabama 35898

August 1982

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898

DISPOSITION INSTRUCTIONS

Destroy this report when it is no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

TRADE NAMES

Use of trade names or manufacturers in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software.

FORWARD

The Logistic Analysis Model LOGAM Technical/Programmer Manual Volume III was written under Contract DAAH)1-82-C-A157. The work was performed with the US Army Missile Command under the general technical cognizance of Mr. Raymon S. Dotson, Systems Analysis Division, Systems Analysis and Evaluation Office, US Army Missile Command, Redstone Arsenal, Alabama. The program also produced two companion documents entitled LOGAM Users Manual Volume II and LOGAM Executive Summary Volume I.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER SR D-82-2	2. GOVT ACCESSION NO. AD-A136 602	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LOGAM Technical/Programmer Manual, Volume III		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DAAH01-82-C-A157
9. PERFORMING ORGANIZATION NAME AND ADDRESS John M. Cockerham & Assoc., Inc. 4717 University Dr., Suite 106 Huntsville, AL 35805		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AMCMS 63310.0740012
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Command ATTN: DRSMI-DS, Redstone Arsenal, AL 35898		12. REPORT DATE August 1982
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Logistics	Maintenance	Systems Analysis
Cost Analysis	Test Equipment	Availability
Operational Support Cost	Manpower	
Supply	Maintenance Policy Evaluation	
Provisioning	Design-Logistics Interface	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This manual describes the Logistic Analysis Model (LOGAM) and its use for evaluating logistic operations as applied to US Army materiel systems. The objective is to develop methodology for generating quantitative data for analysis of activities necessary to equip, operate, maintain, and support a materiel system. LOGAM is a deterministic model, analytical in design through its sensitivity feature, and highly versatile in its ability to evaluate many alternatives rapidly and inexpensively. Through the sensitivity median,

ABSTRACT (Continued)

ABSTRACT (Concluded)

support alternatives are tested for evaluating life cycle costs and for recommending optimum repair levels; repair versus discard at failure; manpower, provisioning and test equipment requirements; table of organization and equipment adjustment or development; and other operational elements by quantities and costs.

Con t'd from page 1

TABLE OF CONTENTS

SECTION	TITLE	PAGE
1.0	Introduction	1-1
2.0	Job Control Language	2-1
3.0	Special Features and Mathematical Formulations	
	3.1 Maintenance Policy Selection	3-1
	3.2 Provisioning Rules	3-5
	3.3 Sensitivity Testing	3-15
	3.4 Availability Accumulator	3-23
4.0	Detailed Program Description	
	Program LOGAM1	4-1
	Block Data BLKDAT	4-107
	Subroutine BASIC	4-109
	Subroutine SUPI	4-116
	Subroutine SENSIT	4-119
	Subroutine PAGE	4-121
	Subroutine IOL	4-122
	Function DEF	4-124
	Subroutine OPER	4-127
	Subroutine ONE	4-130
	Subroutine TWO	4-134
	Subroutine THREE	4-139
	Subroutine FOUR	4-141
	Subroutine FIVE	4-143
	Subroutine SIX	4-144
	Subroutine SEVEN	4-147

Con t'd

TABLE OF CONTENTS CONT'D

SECTION	TITLE	PAGE
4.0	Detailed Program Description cont'd	
	Subroutine EIGHT	4-149
	Function SPOL	4-158
	Function AB	4-160
	Function D	4-161
5.0	Program LOGAM Input Definitions	
	5.1 LOGAM Inputs	5-1
	5.2 Post Processor Inputs	5-26
6.0	Program LOGAM Output Definitions	
	6.1 Individual LRU Outputs	6-1
	6.2 Summarized LRU Case Outputs	6-6
	6.3 Supplementary LRU Outputs	6-11
	6.4 Cost Totals Outputs	6-12
	6.5 Maintenance Outputs	6-15
	6.6 System Support Costs Outputs	6-19
	6.7 Sensitivity Outputs	6-22
7.0	Sample Inputs and	7-1
8.0	Sample Outputs	8-1

FIGURES

FIGURE	TITLE	PAGE
1	Maintenance Policy Matrix	3-2
2	Equipment Level (E)-Maintenance Flow	3-7
3	Direct Support (O) - Maintenance Flow	3-8
4	General Support (I) - Maintenance Flow	3-9
5	Depot Level (D) - Maintenance Flow	3-10
6	Sample Sensitivity Input	3-22

TABLES

TABLE	TITLE	PAGE
1	Policy "G" Factors	3-3
2	Sensitivity Input Listing	3-19
3	Description of Post Processor Line Items	5-26
4	Post Processor Field Input and Mnemonics	5-27
5	Explanation of PP Mnemonics and Use	5-28

SECTION 1

INTRODUCTION

This document is designed to provide sufficient information for understanding the logic and implementation of the LOGAM logistics computer model. LOGAM provides a tool for the evaluation of alternate support postures for Army equipment. The program is structured to perform logistics analyses in maintenance support situations where the emphasis is on the support channels required for a diversity of operating equipments. The following sections are included in this Volume; Page I

The remaining sections of this document and a brief description of each section are as follows:

Section 2 - The control card section, which gives the j control language statements needed to operate the program on MICOM CYBER 74 computer.

Section 3 - The special features and mathematical formulat s, which lists the primary formulas used for evaluating the mainter flows and provisions and a discussion of some important ar of the program.

Section 4 - The detailed program description, which documents the purpose of each statement or small block of statements. The main program as well as each subroutine are described. These descriptions could be useful for future modifications to the program code.

Section 5 - The input definitions, which describes all the inputs used by the program.

Section 6 - The output definitions, which describes the variable names output by the program.

Section 7 - The sample inputs, which is a listing of a case run.

Section 8 - The sample outputs, which document the various pages of output formats. The output pages are annotated to match the variable descriptions of section 7.

Section 9 - The flowchart, which is a computer drawn logic flow using a standard line printer, bound computer output.

Section 10 - The program listing, which is bound computer output.

With a knowledge of logistics and an understanding of the FORTRAN language there should be sufficient information contained in this document to provide an understanding of the program's logic and use. Each section of this document provides a specific type of information; therefore, the sections should be used in conjunction to get a global view of the program.

SECTION 2

JOB CONTROL LANGUAGE

This section discusses the Job Control Language (JCL) required to use the LOGAM program. The JCL here is for a program library that is cataloged on a permanent disk file. The following statements describe the JCL for updating, compiling, and executing the LOGAM program on MICOM's CDC Cyber 74 Computer system.

JOBNO,T77,CM150000.
This statement includes the job number, execution time in octal seconds, and the central memory size, respectively. The job number is a fictitious one, a legitimate one would be provided by computer operations personnel. The time limit is set here because program execution will not complete within the system default time.

ACCOUNT, UN=username, PW=password, AC=accounts, OP=A2, J=0001
This statement is the account information that includes the users name, password, and account number. The letters in this statement that are upper case will always be input. The lower case letters is the information provided by the user. The password and account numbers are provided to the user by computer operations personnel.

LIMIT, 500.
This statement reserves 500 blocks of disk space to store the LOGAM outputs.

ATTACH,OLDPL,Filename,ID=idname, CY=N.
This statement attaches the LOGAM program library from a permanent file into the local file OLDPL for updating later. The information in lower case letters is provided by the user where "filename" must be obtained from computer operations personnel.

UPDATE,F.
This statement performs a full update of the LOGAM program library stored on file OLDPL and provides a COMPILE file for later compilation.

RFL,150000.
This statement sets the central memory field length to 150K octal locations to provide sufficient memory to perform the compilation of the LOGAM FORTRAN code.

FTN,I=COMPILE,R=2.
This statement uses the FORTRAN compiler (FTN) to compile the output file from the UPDATE command to generate an executable program file (LGO).

LGO.
This statement loads and executes the relocatable program file (LGO).

SECTION 3

LOGAM SPECIAL FEATURES AND MATHEMATICAL FORMULATIONS

3.1 Maintenance Policy Selection

The logistic and maintenance support system possibilities which may be considered comprise twenty basic maintenance policies with four possible levels of inventory support for each. The 20 basic maintenance policies are summarized in Figure 1. LOGAM additionally allows the analyst to split maintenance policy and stock location - this leads to a number of combinations which are essentially unlimited.

3.1.1 Policy "G" Factors - The LOGAM deployment matrix shows four possible levels of maintenance support: at the Equipment proper, at a Direct Support level, at a General Support level, and at a Depot level. The model additionally assumes that faults are identified in accordance with the LRU removal rate E at the equipment level. LOGAM also provides three levels of maintenance support capability: unit checkout (COU), fault isolation of the unit to a faulty module (FIM), and module test and repair (FIP). It provides three levels of logistic discard: unit, module, and part.

The maintenance levels at which work is performed and the test equipment, test, and repair manpower locations are specified by "G" factors. These are the same "G" terms illustrated on Figure 1. The same factors are used to define the flow of maintenance work in the postulated system or given deployment. These factors are described in Table 1.

The matrix of the "G" factors as structured to form the maintenance policies that are built into the LOGAM formulation is shown in Figure 1. This matrix identifies the support posture options available within the LOGAM model. These alternatives are designated GA through GT in the upper part of the matrix. Twenty alternatives are available and as discussed previously, they can be combined so that a percentage of work is accomplished by one policy with the balance being accommodated by other policies selected from the matrix. In the matrix, X indicates that the options listed around the perimeter of the chart apply for the block in which the X is located. A blank in a block indicates that there is no applicable action taking place.

For example, the X in the fourth column from the left in the fourth row from the top is to be interpreted in the following way:

> (Start at the left-hand edge of the chart)
>
> For the maintenance policy designated "GD," test equipment will be located at DIRECT SUPPORT. Test equipment at Direct Support can isolate faults to the level of the failed MODULE. Repair will be accomplished by discarding and replacing the failed MODULE.

FIGURE 1
MAINTENANCE POLICY MATRIX

TEST EQUIPMENT WILL BE LOCATED AT / TEST EQUIPMENT CAN ISOLATE FAULTS TO THIS LEVEL / FOR THE MAINTENANCE POLICY DESIGNATED BY

	EQUIPMENT	DIRECT SUPPORT				GENERAL SUPPORT			DEPOT		
	LRU	MODULE	LRU	MODULE	PART	LRU	MODULE	PART	LRU	MODULE	PART
*	TRC	TE TER	TC	TF TFR	TMO TMOR	TC	TI TIR	TMI TMIR	TC	TD TDR	TMD TMDR
GA	X										
GB	X		X								
GC	X	X									
GD	X		X	X							
GE	X					X	X				
GF	X		X			X	X				
GG	X								X	X	
GH	X		X						X	X	
GI	X	X	X	X	X						
GJ	X	X				X	X	X			
GK	X	X							X	X	X
GL	X		X	X	X						
GM	X		X	X		X	X	X			
GN	X		X	X					X	X	X
GO	X		X			X	X	X			
GP	X		X			X	X		X	X	X
GQ	X		X						X	X	X
GR	X					X	X	X			
GS	X					X	X		X	X	X
GT	X								X	X	X

* LOGAM MNEMONIC

LRU / MODULE / PART

REPAIR WILL BE ACCOMPLISHED BY DISCARDING AND REPLACING THE FAILED.

TABLE 1
POLICY "G" FACTORS

GA	Specifies a policy of discard at failure. There are no maintenance support activities. All failures, false "no go" indications, and attrition rate inputs result in LRU discard. Only LRUs are stocked in the supply system. There is no demand for modules or parts.
GB	Similar to GA but here is a provision to detect false "no go's" at Direct Support and only failed and attrited LRUs are discarded. There is no demand for module or part stock.
GC	Specifies LRU repair at equipment level by removing and replacing a defective module. The defective module is discarded.
GD	Specifies LRU repair at Direct Support by removing and replacing a defective module. The defective module is discarded.
GE	Specifies LRU repair at General Support by removing and replacing a defective module. The defective module is discarded.
GF	Specifies LRU repair at General Support with checkout performed at Direct Support to remove false "no go" LRUs before sending the work to General Support. LRU repair is by removal and replacement of a defective module and the defective module is discarded.
GG	Specifies LRU repair at Depot. Defective modules are discarded.
GH	Specifies LRU repair at Depot preceded by a checkout at Direct Support to screen false "no go's". Defective modules are discarded.
GI	Specifies LRU repair at equipment level and module repair at Direct Support.
GJ	Specifies LRU repair at equipment level and module repair at General Support.
GK	Specifies LRU repair at equipment level and module repair at the Depot.
GL	Specifies LRU and module repair at Direct Support.
GM	Specifies LRU repair at Direct Support and module repair

at General Support.

GN Specifies LRU repair at Direct Support and module repair at Depot.

GO Specifies checkout to screen false "no go's" at Direct Support followed by LRU and module repair at General Support.

GP Specifies checkout to screen false "no go's" at Direct Support followed by LRU repaired General Support and module repair at Depot.

GQ Specific LRU checkout to screen false "no go's" at Direct Support followed by LRU and module repair at Depot.

GR Specifies LRU and module repair at General Support.

GS Specifies LRU repair at General Support and module repair at Depot.

GT Specifies LRU and module repair at Depot.

The specification of a maintenance concept, input by the GA through GT fractions, may be mixed in any proportion summing to unity to represent the flow of work demand. For example, if:

GL = 0.6, GR = 0.25, GT = 0.15

60% of the LRU removals would be sent to Direct Support for repair, 25% would be sent to General Support for repair, and the remaining 15% would go to Depot. If FUO = .8 then eighty percent of the total LRUs arriving at Direct Support would be repaired. The other twenty percent would be scrapped since there is no general support in policy GL. Similarly FUI = .8 and FUD = .8 would act for General Support and Depot.

Care should be taken in developing a support concept through the selection of multiple G fractions since the workload flow between maintenance levels may be different even though the concepts developed have the same support levels. For example, lets assume one wants to model a typical 4 level maintenance concept. This could be done by combining either policies GL, GM and GN or GD, GE and GG. Assuming that GL=GD=.6 and GM=GE=.25 and GN=GG=.15 the difference can be highlighted. In the first grouping of policies (GL, GM, and GN) the flow of LRUs from E to DS is equal to (F+FNG)+(GL+GM+GN)=(F+FNG). In the second grouping of policies (GD, GE, and GG) the flow of LRUs from E to DS is equal to (F+FNG)*(GD)=(F+FNG)*.6. Thus the flow of LRUs in the second grouping of policies is .4 less than the first. This is due to the fact that the grouping policies GL, GM, and GN all have a DS support level in the policy matrix while in the grouping of

policies, GD, GE, and GG only the GD policy has a DS support level in the policy matrix. Thus it is critical to understand how the workload flows are structured for a combination of G policies.

3.2 Provisioning Rules

LOGAM has three rules for determining initial provisions. The method of determining the type of rules to use is dictated by the input value for AYZP. A negative value for AYZP will use a predetermined (input) method and the adequacy of the provisions are left for the user to determine. A zero value of AYZP will use the LOGAM supply rule and a positive value will use the maintenance rule.

3.2.1 Predetermined Supply Levels - In this case the analyst inputs the quantity of LRUs (QTE, QTO, QTI, QTD), modules (QTMO, QTMI, QTMD), and parts (QTPO, QTPI, QTPD). The effect that these values have on availability is computed and modified if necessary.

3.2.2 LOGAM Supply Rules - For LRUs QTE is examined first. This represents the modified input value of LRU stock at "E" (equipment) at this point of the program. If its value is greater than 0.5, it is assumed that the input value is valid and the value is the number of LRUs to be stocked. However, if QTE is less than 0.5, it is assumed that this may not be the actual value wanted. H(1), the "stock at E level", is examined to determine if stock is intended to be placed at "E" (it is if H(1)=1). If the answer is yes, the IOL subroutine is called to compute this new QTE value. If no stock is intended to be placed at "E", the QTE value is ignored and in either case the needs for LRU stock at Direct Support, General Support, and Depot are successively explored and, where required, calculated by calling IOL.

3.2.2.1 Module and Part Stock - Module and part requirements are similarly examined and calculated. The logic is similar to that used in the LRU logic, but the "H" flags do not exist for modules and parts and there are only three possible support levels that may stock the modules and parts.

3.2.2.2 IOL Operation - The IOL subroutine is called with the safety stock coefficient (CKK), the number of locations (XD), the quantity tied up in scrap replenishment (BQU), the quantity tied up in float repairs (BQF) (this is zero for parts), and the round-up point (Z) known. The IOL subroutine adds the two tied-up quantities and multiplies by the number of possible locations (XD). The output (BQT) is set at zero and routine checks on XD and QUF are made for obvious errors in which case the subroutine returns with the output stock quantity (BQT) at zero.

The safety stock increment (BSQ, the standard deviation from the mean or expected value of stock tied up) is computed as the square root of the mean (per the Poisson distribution).

The amount of safety stock is computed by multiplying the safety stock increment (BSQ) by the coefficient (CKK), an input quantity. This is given the name BQS.

The total stock is computed by adding the pipeline stock (BQU), the repair float stock (BQF), and the safety stock (BQS). It is given the name QUFS and is permitted to have a minimum value of zero.

The total stock (QUFS) is divided by XD to obtain the amount of stock per location, the round-up quantity (Z) is added, and the result is integerized to obtain an integer quantity per location. This integer quantity is multiplied by the number of locations (XD) to obtain total stock quantity of LRUs, modules, or parts. Because of the rounding-up process, the resulting stock quantity always exceeds the expected value of demand for stock.

3.2.3 LOGAM Maintenance

LOGAM Maintenance Rules allow for determination of initial provisioning quantities of LRU's, modules, and parts based upon time delays due to the maintenance concept or concepts selected. Figures 2 through 5 depict the logic used to generate the repaired, screened, evacuated and scrap flows associated with the maintenance concepts.

3.2.3.1 Computation of LRUs, Modules, and Parts - In order to fully understand the LOGAM maintenance rule, it is desirable to discuss the LRU flows through the various maintenance levels.

At the equipment level (ED) provisioning is a function of the failure rate (E) and:

1) Mean time to repair (TRC). It is the total corrective maintenance down time in clock hours during the stated period to return the materiel system to an operational state. TRC must be transposed into days for computation of stock. TRC will always have a number.

2) Mean time in days spent awaiting a replacement LRU or module (TATE). It is the average length of time in days required to obtain an LRU or module (based upon the maintenance policy) from a support activity. TATE must be calculated based upon the authorized location of replacement LRUs and modules as expressed in the H(1) matrix.

FIGURE 2
EQUIPMENT LEVEL (E) - MAINTENANCE FLOW

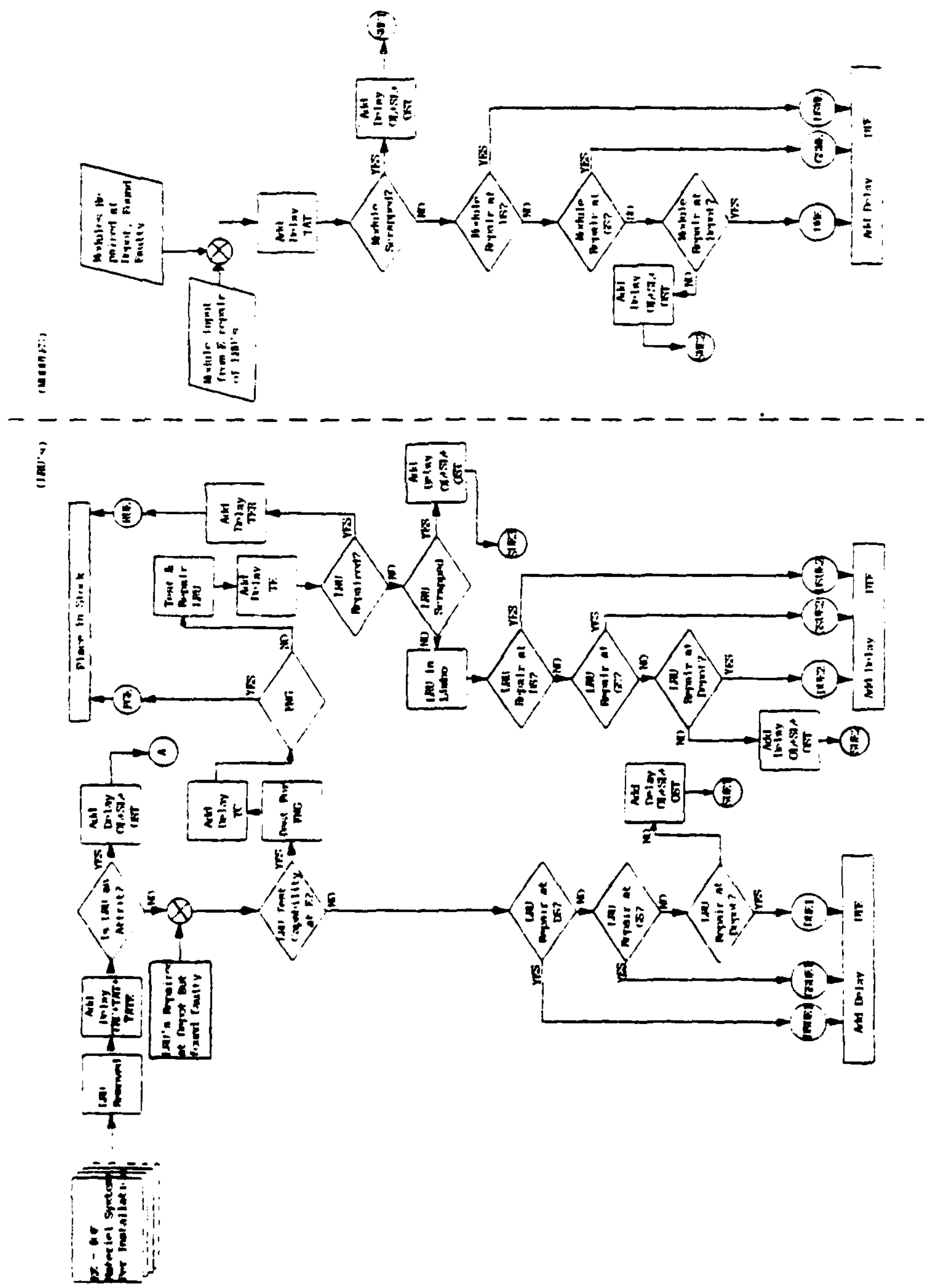


FIGURE 3
DIRECT SUPPORT LEVEL (O) - MAINTENANCE FLOW

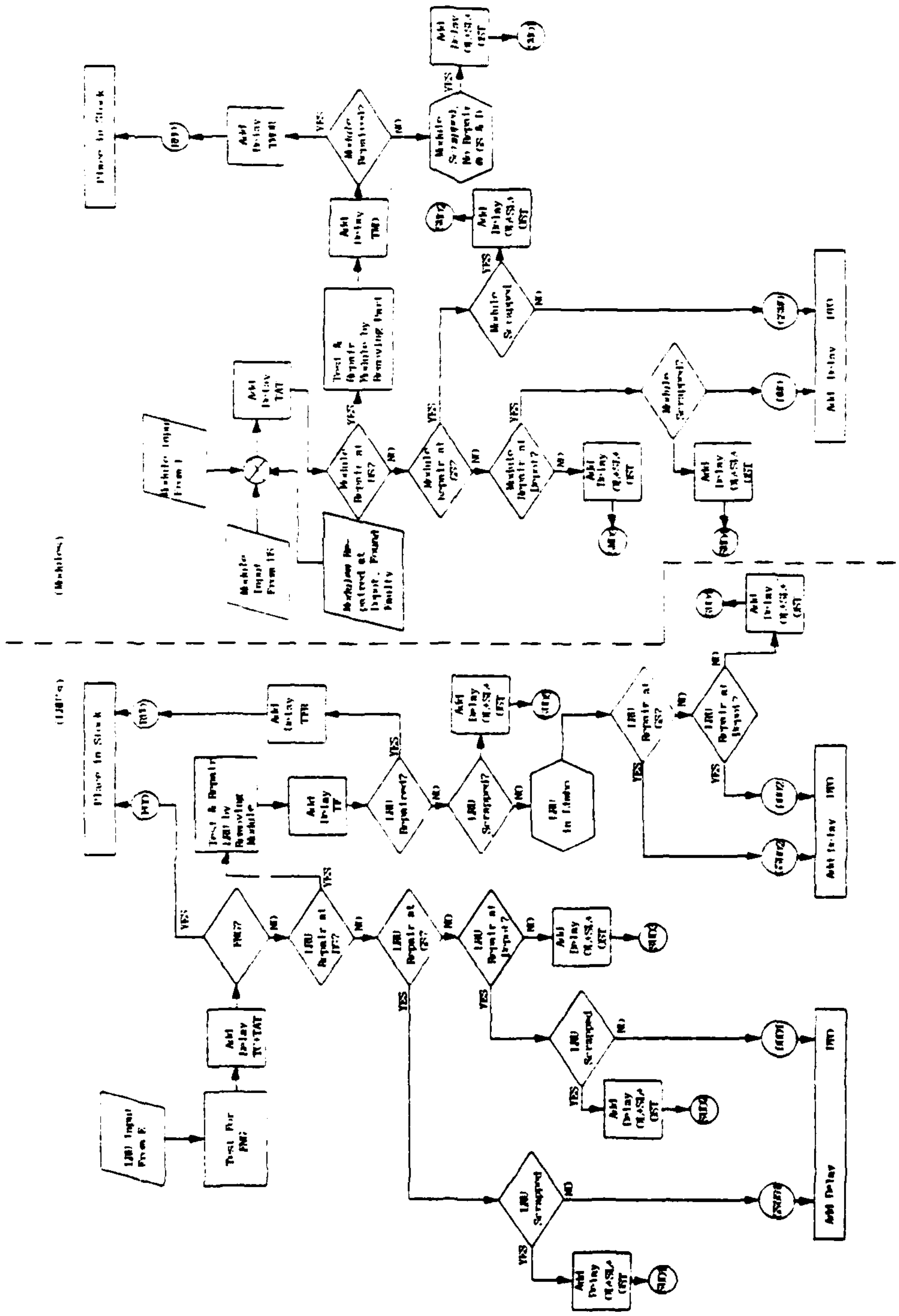


FIGURE 4
GENERAL SUPPORT LEVEL (I) - MAINTENANCE FLOW

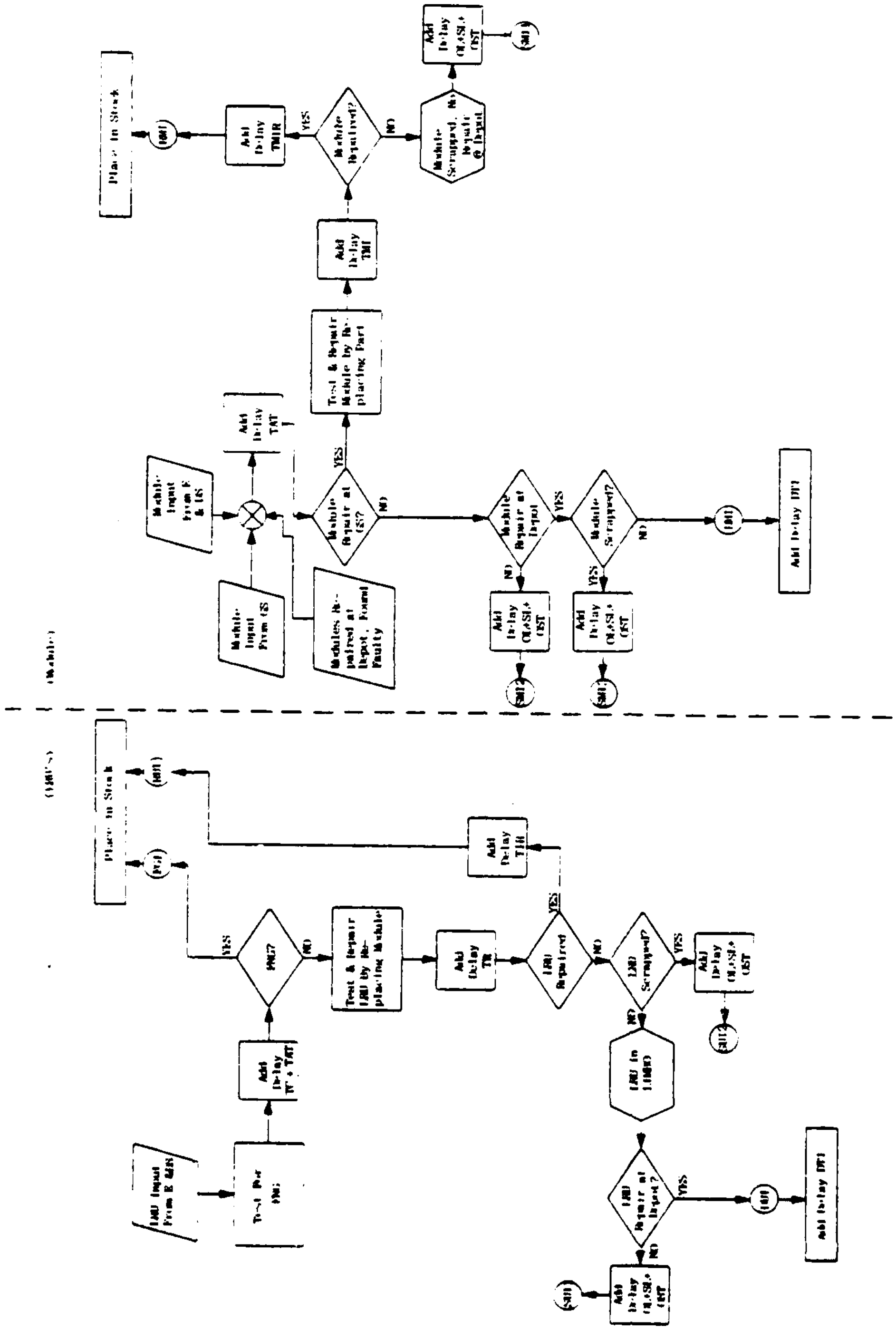
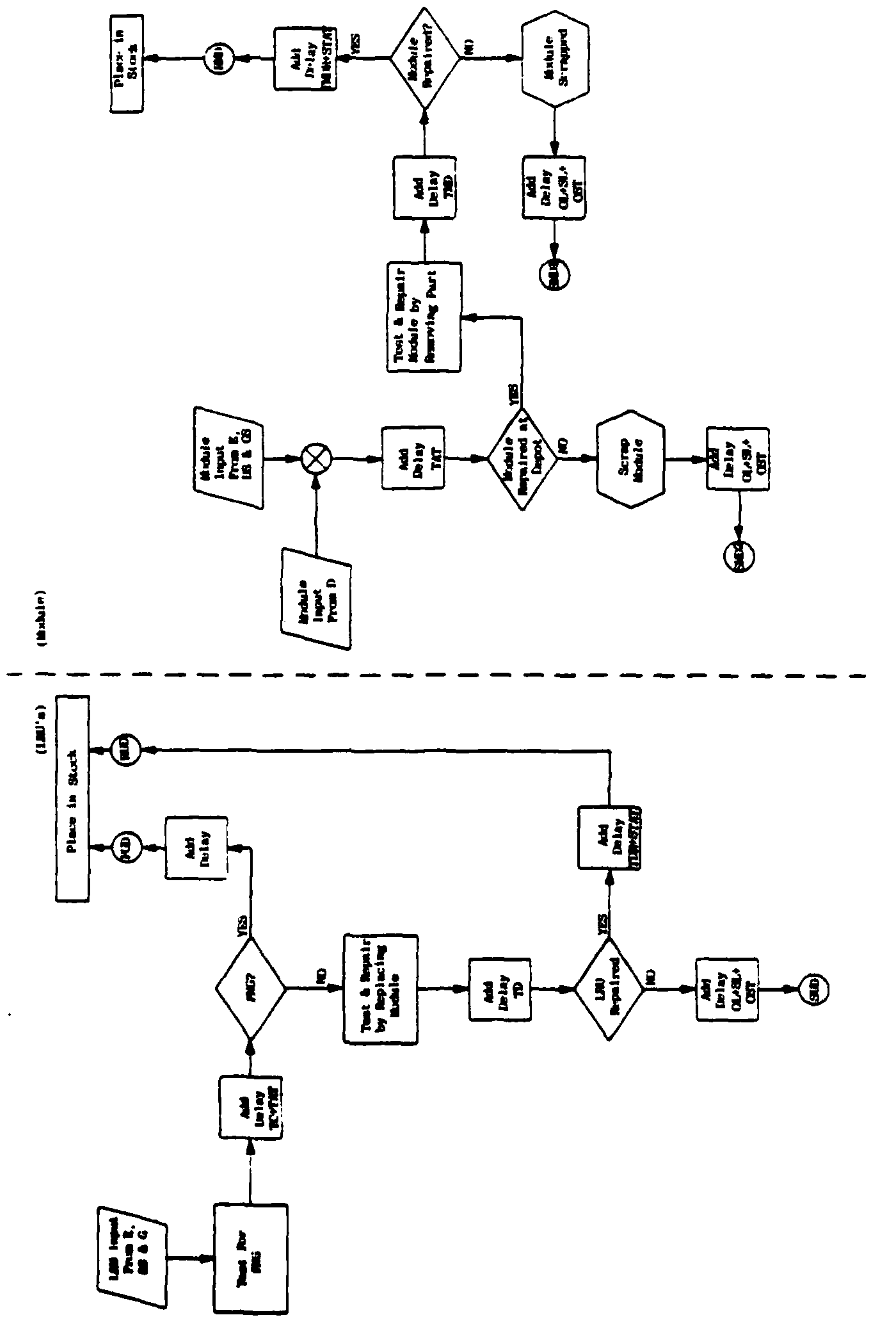


FIGURE 5
DEPOT LEVEL (D) - MAINTENANCE FLOW



3) Mean downtime in days awaiting maintenance personnel, shop space or transportation at ED (TAT(1)).

4) Mean time in days an LRU or module awaits evacuation to the next maintenance echelon (DTE). This is to account for waiting time to package and ship an item. If a maintenance policy calls for all LRUs to be evacuated, DTE is charged to all LRUs. If a policy calls for a percent of LRUs to be repaired at ED and a percent to be evacuated, DTE is charged to the percent evacuated.

5) Average time in hours to check for a false "no go" (TC). It must be transposed to days for computation of stock. Only LRUs are checked for false "no go's" and only in maintenance concepts that provide for LRU repair/checkout in the Maintenance Policy Matrix. Only policies GC, GI, GJ, and GK have LRU screening capability at ED Level.

6) Mean time in hours to test a LRU at ED level (TE). It must be transposed to days for computation of stock.

7) Mean time in hours to repair an LRU at ED level (TER). It must be transposed to days for computation of stocks.

8) Operating Level, Safety Level and Order and Ship Time (OL+SL+OST) in days. This delay is used to account for stocks to replace those that are scrapped.

Provisioning quantities required to support the maintenance policy are computed considering the flow of LRUs and modules through the selected maintenance concept. As the LRU or module flows through the system provisioning days are computed and accumulated based upon the delays encountered. For example an LRU removed at ED and evacuated to DD would accumulate delays TRC, TAT(1), TATE & DTE while an LRU which was removed, repaired, and returned to stock at ED would accumulate delays TRC, TAT(1), TATE, TC, TE and TER. Thus as the LRUs and modules flow through the evacuate, screen for false "no go", repair, and scrap gates the delays associated with these actions are accumulated and initial provisions of LRUs and modules are computed to account for the time required to process (repair, evacuate, etc.) the unserviceable LRU or module. The following calculations are made to compute LRU provisioning quantities at ED level. (See Figure 2).

Since all LRU removals are charged TRC, TAT(1) and TATE delays, the delays are computed at time of removal rather that at time of disposition (e.g. evacuate, repair, scrap & screen). PDRE is the provisioning days required to account for removal of LRUs at Equipment level.

PDRE = TU*(TRC/(24*WER/168)+TAT(1)+TATE)

where: TU=(F+FNG+A) is the total number of LRUs removed from the material system.

TRC is divided by 24*WER/168 to translate TRC to days and to allow the work week in hours (WER) of ED level maintenance personnel to be considered.

****

Evacuation of LRUs to higher support levels causes delays depending on how they are evacuated. They can be evacuated immediately because ED level has no LRU repair capability or they can be evacuated after repair was attempted but did not succeed and the LRU was placed in the "in limbo" status. PDEUE1 is the provisioning days for those LRUs for which the G policy allows no repair at ED level. The evacuated flow to OD, DI, and DD are identified as DSUE1, GSUE1, and DUE1.

```
DSUE1=(F+FNG)*(GB+GD+GF+GL+GM+GN+GO+GP)+
      F*(DPT)*( GH+GQ)+FNG*(GH+GQ)
GSUE1=(F+FNG)*(GE+GR+GS)
 DUE1=F*DPT*(GE+GT)+FNG*(GG+GT)
```

where: DPT=2.-DAOQL is used to compute the Depot work/load recycled.

since all policies except GC, GI, GJ, and GK have false "no go's" screening capability all the true failures plus false "no go's" must be evacuated to the next level. The term DPT accounts for LRUs that were sent from ED to DD level (policies GH, GG, GO and GJ) that were supposedly fixed but when installed in the material system were found faulty.

The delay is calculated by multiplying the total flow by the handling delay at equipment level (DTE).

```
PDEUE1=(DSUE1+GSUE1+DUE1)*(DTE)
```

PDEUE2 is the provisioning days for those LRUs for which the G policy allows repair at ED level. Since not all LRUs are repaired due to the FUE fraction some end up "in limbo" and are evacuated. This percentage is identified as ULE.

```
DSUE2=F*ULE*(GI)
GSUE2=F*ULE*(GJ)
DUE2=F*ULE*(GK)
```

The total provisioning days are then calculated by multipling the total flow by the handling delay (DTE), the time to screen the LRU for a false "no go" (TC) and the time to test the LRU (TE).

```
PDEUE2=(DSUE2+GSUE2+DUE2)*((TE+TC)/(24*WER/168)+DTE)
```

The PDEUE is the total number of provisioning days at ED level due to

evacuated LRUs and is computed by summing the two delays.

PDEUE=PDUE1+PDUE2

****

The scrapping of LRUs generates provisioning days depending on how they were scrapped. They can be scrapped because for whatever reason, the LRU could not be repaired at ED level and the G policy had no higher level authorized (OD, DI, or DD) or because of items that were declared not economically repairable and are then discarded at ED level. Those units which are scrapped due to a G policy with no higher maintenance level are charged only Operating Level, Safety Level, and Order and Ship Time (OL+SL+OST) delays, since these items were not screened and tested before they were scrapped. All other scrap will have the additional charge for FNG screening (TC) and test time (TE).

The various scrap flows are identified as SUE1, SUE2, and SUE3.

SUE1=(F+FNG)*GA
SUE2=F*ULE*GC
SUE3=F*SUE*(GC+GI+GJ+GK)

PDSUE=(SUE1+SUE2+SUE3)*(OL(1)+SL(1)+OST(1))+
(SUE2+SUE3)*(TC+TE)/(24*WER/168)

****

Attritions add additional provisioning days, in that stock must be provided to account for the loss of the LRU. Attritions are those LRUS which are removed from the materiel system at ED level but for whatever reason (lost, stolen, etc.) are lost to the supply system. The provisioning delay due to attritions is similar to scrap

PDAU=A*(OL(1)+SL(1)+OST(1))

****

The time to test and identify false "no go" LRUs also contributes to provisioning days and is charged a time for testing (TC). At ED level, false "no go" screening capability is available only for concepts GC, GI, GJ, and GK. The flow of false "no go's" which are screened at ED level and found serviceable and returned to stock is identified as FGE.

FGE=FNG*(GC+GI+GJ+GK)

The number of provisioning days is computed by multiplying the flow and the FNG screening (TC) time.

PDSKUE=FGE*(TC/(24*WER/168)

****

The LRU repair flow at ED level is expressed as RUE.

RUE=F*URE*(GC+GI+GJ+GK)

The provisioning days due to repair (PDRUE) are computed by multiplying the flow by delay times. LRUs which have been repaired have gone through false "no go" screening and are charged TC. They also have undergone diagnostic testing and are charged TE, and finally they have been repaired and are charged repair time TER.

PDRUE=RUE*(TC+TE+TER)/(24*WER/168)

****

Since provisioning is based upon the system operating hours, operational availability and the total number of systems deployed it is necessary to multiply PDUE by these factors and convert from hours to days. Thus the term OFACT is computed by multiplying the operating time factor (OTF) by 8766 hours per year by the number of systems per installation (ED) and by the inherent availability (AYZ) and then dividing by 365 days per year.

OFACT OTF*8766*ED*AYZ/365

Thus the total provisioning quantity for LRU's at ED level is:

PDUE=PDUE*OFACT

The remainder of the calculations for LRUs, modules, and parts follow the same logic as shown above and will not be duplicated.

****

3.2.3.3 <u>Distribution of LRUs, Modules, and Parts</u> - The equations used to distribute the required LRUs based upon the stockage matrix (H(I)) are designed to distribute spares to the lowest support level (ED) based upon the H array. For example, assume the H array is H=1,1,1,1, which places stock at each echelon and EDS=6, ODS=2, DI=1 and DD=1. Also assume that the required provisioning quantities computed as explained above are as follows:

MAINT. LEVEL	LRU QTY
ED	12.72

OD	7.95
DI	4.77
DD	27.56

In allocating LRUs, a round-up of .5 is assumed. Since the ED LRU requirement is 12.72 and EDS=6 we can have 2 LRUs at each EDS location. The OD LRU requirement is 7.95 and a LRU was left over at EDS due to round off. This added to 7.95 for ODS equals 8.67. The round off is to 10; therefore, each ODS location will have 5 LRUs. The DI LRU distribution is 4.77 but 1.33 LRU extra was located at OD; therefore, the single DIS location will require 3.44 but the round off is to 3. The DDS location will have the remaining 27. The distribution of provisioning is as much as possible, determined by the requirement at each echelon.

In all cases the H array controls stock location. If stock is not located at an echelon, but work is performed there, the amount of stock that would be computed for that location is added to the next more forward echelon i.e., if H=1,0,1,1, and LRU stock is computed as shown for the sample case. ED=12.72, OD=7.95, DI=4.77, and DD=27.56, stock is located as follows:

ED=20.67, OD=0, DI=4.77, DD=27.56
Rounded to ED=24, OD=0, DI=5, DD=24

If H=0,0,1,1, stock would be located at the forward most echelon authorized stock. Stock is located as follows:

ED=0, OD=0, DI=25.44, DD=27.56
Rounded to ED=0, OD=0, DI=25, DD=28

In cases where stock is located at a higher echelon the analyst should assure that the mnemonic TATE expresses a reasonable time to obtain the needed LRU or module.

The logic for module and part distribution is also tied to the H array except that parts are not stocked at ED.

3.3 Sensitivity Testing

After a set of individual LRU cases have been run as a baseline case, it is often desirable to be able to rerun the entire data set with selected changes in certain of the input variables. To facilitate this, the program writes a copy of the input data to a memory device during the baseline run. Subsequently, these data may be retrieved, edited, and rerun. These reruns of the input based on selected editing are referred to as sensitivity runs.

3.3.1 Sensitivity Input Array - One of the elements of the input NAMELIST/L/ is an array named SENSY. Values input to this array are used to direct the conduct of sensitivity runs. The array SENSY, stored in common block SENS, has Dimension 266. Entries into these 266 storage locations perform the following functions:

a) Specify the number of input variables whose values are to be edited during the sensitivity runs.

b) Specify the number of times the inputs are to have their values edited. (This specifies the number of sensitivity runs).

c) Specify the rules to be used for the editing of each designated input.

d) Designate the inputs to be altered.

e) Furnish the numeric values to be used by the specified rules in the edition of the designated inputs.

3.3.1.1 First Element of the SENSY Array - In more detail, the first element of SENSY, i.e., SENSY(1), is used to accomplish Function (a) in Section 3.3.1. A positive, real, whole number is entered to state the number of inputs being tested. Within the program, this is called MODE. This program is currently written so that MODE may range from one to twelve inputs. More than twelve inputs results in an error message:

BAD SENSY

followed by a printout of the contents of array SENSY, the sensitivity test is abandoned, and the program resumes as though it were a new start after completing sensitivity testing.

The exact value 0 is used to denote that sensitivity testing is off and the program is running baseline cases. This value exists at program start by initialization in a BLOCK DATA subprogram. Thus, SENSY need not be input to run the baseline case. Similarly, after the completion of all the work of a sensitivity run, SENSY(1) is reset to zero and no input is needed to resume analysis of baseline cases. In fact, all elements of SENSY are reset to zero. (Input of negative values in SENSY(1) are not detected by the program. The program will run SENSY with unpredictable results. Negative values should not be entered for SENSY(1).

3.3.1.2 Second Element of the SENSY Array - The second element of SENSY, i.e., SENSY(2), is used to carry out Function (b) given in Section 3.3.1. A positive, real, whole number is entered to stipulate

the number of sensitivity runs. This is known as NPASS within the program. Due to the limitations of the dimensionality of SENSY, there is a limit to the number of passes that can be made by one loading of SENSY. The number depends on MODE. The limits on NPASS for the twelve possible values of MODE is listed below.

MODE	NPASS LIMIT
1	262
2	130
3	86
4	64
5	50
6	42
7	35
8	31
9	27
10	24
11	22
12	20

The remaining elements of SENSY are furnished as ordered sets of size MODE. Thus, if only one input is being tested, the set size is one; if two, the size is two, etc. up to the limit of twelve per set when MODE is 12.

3.3.1.3 <u>Third Element of the SENSY Array</u> - Function (c) in Section 3.3.1 is the specifications of the edited rules. This is accomplished by furnishing a set of positive, real, whole numbers. There is one rule number in the set for each of the MODE variables to be varied. The permissible rule numbers are as follows:

Rule Number	Effect
1.	Assign
2.	Add
3.	Subtract
4.	Multiply
5.	Divide

If any other value is used, an error message will be written as follows:

ILLEGAL RULE KRULE = X

giving the sequence of the rule. That input will not be altered and the program will continue. Later sets of entries in SENSY contain values to be used with these rules. Thus, for Rule 1, the value furnished is used instead of the value in the baseline data. Rules 2, 3, 4, and 5 take the value using addition, subtraction, multiplication, or division as specified.

Within the program, the set of rules is stored in array NRULE, of Dimension 12. Should Rule 5 ever encounter the value zero in SENSY, the error message

ATTEMPTED DIVIDE ERROR INDEX = X

will be written where X will be the sequence number in the SENSY array. The program will continue using the baseline value for that variable.

Thus, with MODE in SENSY(1), NPASS in SENSY(2), the set of MODE rules are entered in SENSY(3) to SENSY (MODE + 2).

3.3.1.4 <u>Designation of the Variables to be Tested</u> - In the designation of the variables for sensitivity testing, the program is structured to reference them by their numbered positional location in common block INPUT rather than by name. The numbered sequence for addressing LOGAM inputs to be sensitivity tested is given in Table 2. The sequence numbers are also included with the input definitions in section 5.1. (INPUT DEFINITIONS). Thus, to refer to input E, the LRU failure rate, the number to be entered in SENSY is 81. The reference numbers are to be entered as positive real whole numbers. Should a value other than those in the table be entered, an error message will be entered as follows:

ILLEGAL VARIABLE ADDRESSED = M

where M is the illegal number. The program will continue and no variable will be altered for that bad value.

Thus, to carry out Function (d) in Section 3.3.1, an altered set of MODE variable numbers is entered into SENSY (2 + MODE + 1) through SENSY (2 + MODE + MODE). These are stored in the program array NVAR, of Dimension 12.

3.3.1.5 <u>Designation of the New Values for the Inputs</u> - The remaining portion of SENSY is used to enter NPASS sets of MODE elements to carry out Function (e) in 3.3.1, i.e., supply the values to be used to alter each variable designated, according to the rule, for each pass. Thus, to recapitulate:

SENSY(1)	MODE	Number of inputs to be tested.
SENSY(2)	NPASS	Number of Runs or Passes.

TABLE 2
SENSITIVITY INPUT LISTING

1	ARA	41	CKPO	81	E	121	OTF
2	AYZP	42	CKUD	82	ED	122	P
3	CAD	43	CKUE	83	EDS	123	PMR
4	CALMAN	44	CKUI	84	EE	124	PP
5	CALPUB	45	CKUO	85	EVDM	125	PRP
6	CALSET	46	CLRUPG	86	EVDR	126	PUR
7	CCAL	47	CMODPO	87	EVDT	127	MM
8	CCALP	48	CMP	88	EVEM	128	QMP
9	CCALR	49	CONMAN	89	EVER	129	QMU
10	CCSP	50	CONTCT	90	EVET	130	QTD
11	CCSPP	51	CPE	91	EVIM	131	QTE
12	CCSPR	52	CPI	92	EVIR	132	QTI
13	CDDI	53	CPII	93	EVIT	133	QTMD
14	CDEO	54	CPP	94	EVOM	134	QTME
15	CDFD	55	CPUBII	95	EVOR	135	QTMI
16	CDID	56	CRI	96	EVOT	136	QTMO
17	CDIO	57	CRII	97	FI	137	QTO
18	CDIST	58	CRM	98	FII	138	QTPD
19	CDMAN	59	CRP	99	FINT	139	QTPI
20	CDOE	60	CRU	100	FMD	140	QTPO
21	CDOI	61	CSDEP	101	FMI	141	RDD
22	GDPMAN	62	CSDSU	102	FMO	142	REPEAT
23	CDPRMN	63	CSESU	103	FN	143	RID
24	CDRMAN	64	CSGSU	104	FNGF	144	ROI
25	CEMAN	65	CTCPUB	105	FNSP	145	SMD
26	CEN	66	CTRA	106	FSA	146	SME
27	CEND	67	CTRCAL	107	FTI	147	SMF
28	CERMAN	68	CTRI	108	FTII	148	SMI
29	CFTD	69	CTRII	109	FTM	149	SMO
30	CGMAN	70	CTRSPT	110	FTP	150	SPE
31	CGRMAN	71	CUBEM	111	FTU	151	SPEV
32	CI	72	CUBEP	112	FUD	152	SPEVR
33	CII	73	CUBEU	113	FUE	153	SUD
34	CKIT	74	CUCE	114	FUI	154	SUE
35	CKMD	75	CUP	115	FUO	155	SUI
36	CKME	76	DAOQL	116	HPM	156	SUO
37	CKMI	77	DD	117	HPP	157	SVE
38	CKMO	78	DDS	118	HPU	158	SVR
39	CKPD	79	DI	119	OD	159	SVT
40	CKPI	80	DIS	120	ODS	160	SVV

TABLE 2 (CONTINUED)
SENSITIVITY INPUT LISTING

161	TALMAN	201	TRC	241	OST(4)
162	TATE	202	TUMD	242	SL(1)
163	TC	203	TUMI	243	SL(2)
164	TD	204	TUMO	244	SL(3)
165	TDI	205	WD	245	SL(4)
166	TDMAN	206	WI	246	TAT(1)
167	TDMW	207	WDR	247	TAT(2)
168	TDPMI	208	WE	248	TAT(3)
169	TDPMII	209	WEM	249	TAT(4)
170	TDPRI	210	WER	250	TAYZ(1)
171	TDPRII	211	WI	251	TAYZ(2)
172	TDR	212	WIM	252	TAYZ(3)
173	TDRMAN	213	WIR	253	TAYZ(4)
174	TE	214	WM	254	TAYZ(5)
175	TER	215	WO	255	TAYZ(6)
176	TEMAN	216	WOM	256	TAYZ(7)
177	TERMAN	217	WOR	257	TAYZ(8)
178	TEO	218	WP	258	TAYZ(9)
179	TF	219	WTKIT	259	TAYZ(10)
180	TFR	220	WU	260	ZM(1)
181	TGMAN	221	YAT	261	ZM(2)
182	TGRMAN	222	YD	262	ZM(3)
183	TI	223	YMWO	263	ZM(4)
184	TID	224	YP	264	ZP(1)
185	TIMW	225	YR	265	ZP(2)
186	T10	226	YZ	266	ZP(3)
187	TIR	227	ZFL	267	ZU(1)
188	TMD	228	ZI	268	ZU(2)
189	TMDD	229	ZO	269	ZU(3)
190	TMDR	230	H(1)	270	ZU(4)
191	TMI	231	H(2)	271	STAT
192	TMID	232	H(3)	272	DTE
193	TMIR	233	H(4)	273	DTO
194	TMO	234	OL(1)	274	DTI
195	TMOD	235	OL(2)	320	REO
196	TMOR	236	OL(3)	321	ARAD
197	TOE	237	OL(4)	322	CTRAD
198	TOI	238	OST(1)	325	TENMAN
199	TOMW	239	OST(2)		
200	TONMAN	240	OST(3)		

SENSY(3) to MODE + 2	NRULE	Set of Rules for Editing.
SENSY [(MODE + 3) to (MODE + MODE + 2)]	NVAR	Designation of Input Variables.
SENSY [(MODE + MODE + 3) to (MODE + MODE + MODE + 2)]		First Set of Variables and so forth.

3.3.2 Example of NAMELIST Inputs for Typical Sensitivity Run

If it is desired to investigate the simultaneous variation of failure rate and false "no go" fraction, a typical set of values would be as follows:

MODE = 2.

To run three sets of data:

NPASS = 3.

The input designators, from Table 2

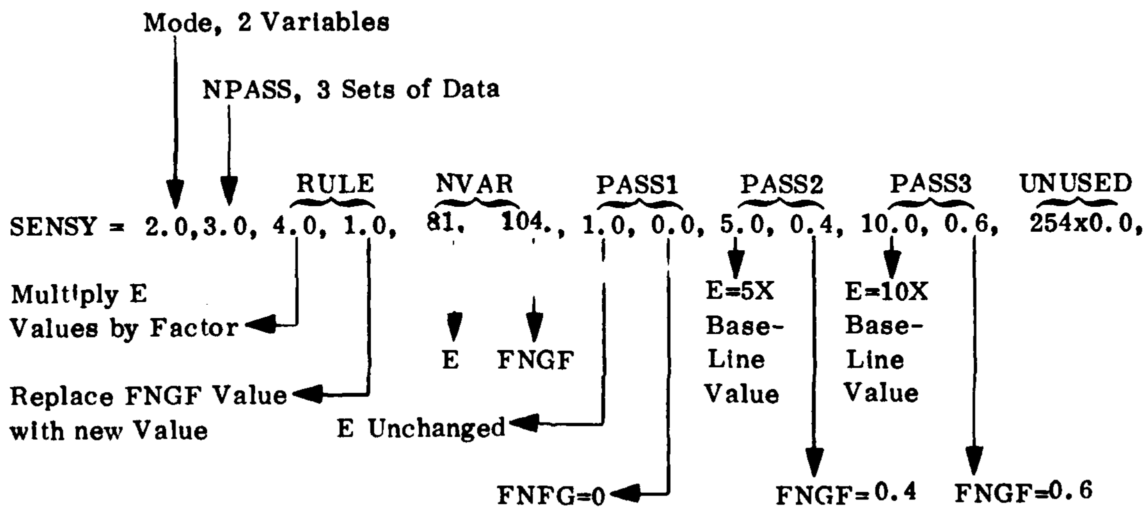
E 81
FNGF 104

In the baseline run, if FNGF was 0.2 for all LRUs and it is desired to run 0., 0.4, and 0.6, for all LRUs, then Rule 1 is used and 0., 0.4, and 0.6 are assigned at the first, second, and third pass, respectively.

For the failure rate E, all LRUs have different values. Rule 1 is not useful as there is no desire to assign the same failure rate to each LRU. More commonly, it is desirable to run multiples of the baseline. Thus, Rule 4 is useful. If no change is desired for the first pass, then the value 1 is used. (This will show the effect of FNGF = 0, without changing E). If for second and third passes, simultaneously with the doubling and tripling of the false "no go" fraction, it is desired to increase the failure rate by factors of 5 and 10, then values 5 and 10 are used. The data input via NAMELIST/L/ for this sensitivity case will have the appearance as shown in Figure 6.

The program will run each pass specified by SENSY. The baseline data store is "rewound" after each pass. The stored baseline data are not ALTERED. After the last pass, control returns to the program and execution continues looking for new input data. At this point,

FIGURE 6
SAMPLE SENSITIVITY INPUT



another SENSY may be entered and the same saved baseline data will be further sensitivity tested.

If another SENSY is not entered, the new baseline cases may be entered. In such a case, the old saved baseline data are destroyed and the new set is saved. At this point, a control (NU=-4) may be entered to stop the program.

3.3.3 Sensitivity Testing Specification - Included at the end of Section 7.1 is a sensitivity NAMELIST input data set that was run with the baseline USAREUR and CONUS data set. As shown, the input cards for sensitivity testing are placed after the final LRU data set. To use the sensitivity test feature of LOGAM, at least three cards must be punched. The first two are generally used as header cards to identify certain factors pertaining to the particular sensitivity run set. The third (and subsequent cards if required, the exact number depending on the number of variables and passes to run) is the input array in the NAMELIST format as discussed in Section 5.1.2.

3.3.4 Sensitivity Output - Along with the SENSY array, the control INHIB may be used to suppress the individual LRU printout. If INHIB = 1 is used, the output page for the final LRU and the totals page only will be printed. If the control INHIB is not activated, the output printout contains the same number of pages as the baseline results. These printouts, however, will show the results for the new values of the inputs as controlled by the rules contained in the SENSY NAMELIST array.

The printout of output on totals pages will always be preceded by a printout of the new values of the inputs identified by the designation number given in Section 5.1.2. Thus, the new value of the input assigned by activating the sensitivity test feature of LOGAM is always documented. Sample output using SENSY data is discussed in Section 6.10 and 6.11.

3.4 Availability Accumulator

Inherent and operational availabilities can be accumulated at the system and subsystem levels. System availability is the accumulation for all LRUs in a case total and subsystem availability would be for groups of LRUs that make up a case total. The groupings of LRUs are determined by the inputs for TAYZ. When the inputs for TAYZ are such that accumulations are made, the accumulated values are stored into arrays CAYZI and CAYZ. CAYZI and CAYZ are the inherent and operational availability arrays, respectively. The first location of each array will contain the system availability and each subsequent NA locations contain subsystem availabilities. NA is input as the number of accumulations to be made; i.e., the system plus the number of subsystems. The following discussion includes an example for availability accumulations.

TAYZ is an array of dimension ten to provide the capacity for ten availability accumulators (definition for NA specifies how many of the ten accumulators are active). A value must be entered for each of the ten availability accumulators; however, only the first NA of the ten are actually used. For example, if a system consists of eleven LRUs and if that system logically subdivides into functional subsystems, the arrangement of the LRUs in the input data deck should be the first four LRUs that constitute the first subsystem, the next five constitute the second subsystem, and the last two constitute the third subsystem. Then if the user wanted to keep the availability tally for the total system and also for each subsystem, four tallies are required. He would input NA = 4. For TAYZ, he would input the following:

LRU No.		LRU No.	
1	TAYZ = 1., 1., 8*0.,	5	TAYZ = 1., 0., 1., 7*0.,
2		6	
3		7	
4		8	
		9	
		10	TAYZ = 1., 2*0., 7*1.,
		11	

All LRUs would be tallied into the first accumulator, i.e., the first element of the TAYZ array is unity for every LRU. The first four LRUs would be tallied into the second accumulator, i.e., the second element of TAYZ is unity for the first four LRUs and zero for all others. LRUs five through nine would be tallied into the third accumulator, i.e., the third element of TAYZ is unity for these LRUs and not for any others. The last two LRUs will be tallied into the fourth accumulator, i.e., the fourth element of TAYZ is unity for these two and zero for all others. Values of TAYZ beyond the fourth element are immaterial because NA = 4. On the case total page, four availabilities will print across the page. The first will be the system availability. The second will be the availability of the first subsystem. The third will be for the second subsystem. The fourth, and last, will be for the third subsystem. Availability outputs are discussed in section 6.5, Maintenance Outputs.

SECTION 4

DETAILED PROGRAM DESCRIPTION

A. This section of the LOGAM1 program assigns computer memory locations to input variables and to those variables that are evaluated in the program. These assignments are made by using labled COMMON, DIMENSION, and EQUIVALENCE statements.

1. PROGRAM LOGAM1 (INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,
TAPE1,TAPE17,TAPE18,TAPE19)
This is the program statement for the main routine. The files used in the program are defined by this statement. TAPE5 is the input file from which all data inputs are read. TAPE6 is the output file from which all results are printed. TAPE17 and TAPE18 are files for storing summarized LRU results. TAPE1 and TAPE19 are used to store the individual LRU data inputs.

2. COMMON/POUT/POUT(13),AMULT,OPERSV,OPMNCS
COMMON/T/T(2000)
COMMON/OFF/OFF(6)
COMMON/ENLM/ENLM(7)
These common blocks are used in Subroutine OPER and the eight subroutines called by OPER to compute and print system support results. They are also used in the BLOCK DATA routine BLKDAT to initialize the variable lists.

3. COMMON/HEADER/DATE(3),TEXT(48),ANALYSIS(3),UNITIS(5),
REMARK(12),COSTIS(6),IPAGE
This common block contains the information to be printed as the header on each page of output. It is used in subroutine EIGHT and PAGE.

4. COMMON/BAS/QFMO,F,SMOC,BBMI,BBMD,QFMI,SMIC,BO,
SUOC,QFMD,BI,QMO,TSMO,QMI,TSMI,QMD,TSMD,
QMDH,QPO,TSPO,QPI,TSPI,QPD,TSPD,QPDH,QUO,
TSU,QUI,QUD,QUDH,OO(20),QI(20),QE(20),QFO,
QFI,QFD,RO(20),RI(20),RD(20),TRI,TRD,QYE,
QYO,QYI,QYD,QYF,TU,QUE,QFE,TMFO,TMFI,TMFD,
TSME,TUFTE,TUFTO,TUFTI,TUFTO,TMFEO,TMFOI.
TMFID,QFME,QME
This common block is used in Subroutine BASIC when values computed in LOGAM1 are needed to compute shipping costs and pipeline quantities.

5. COMMON/BS/X1,X2,X3,X4,X5,X6,X7
This common block is used in Subroutine EIGHT to print the error check for the PAM cost equations of LOGAM1.

6. COMMON/DAPAM/UVA(16)
This common block is used in Subroutine EIGHT to print System Maintenance and Operating Support costs.

7. DIMENSION SDA(16)
EQUIVALENCE(SDA(1),F1),(SDA(2),F2),(SDA(3),F3),

```
        (SDA(4),F4)
EQUIVALENCE(SDA(5),F5),(SDA(6),F6),(SDA(7),F7),
        (SDA(8),F8)
EQUIVALENCE(SDA(9),F9),(SDA(10),F10),(SDA(11),
        F11),(SDA(12),F12)
EQUIVALENCE(SDA(13),F13),(SDA(14),F14),
        (SDA(15),F15),
EQUIVALENCE(SDA(16),F16)
```

These statements dimension the array for accumulating system maintenance and operating support costs, and equivalences each element in the array to an "F" location. The maintenance support costs in LOGAM1 are accumulated into F1 through F16.

8.
```
COMMON/ZERO/CCET,CCTS,CCTSR,CCF,CCM,CCMF,CCMD,
        CCMFD,CTRF,CTRDEP,CTR,CIV,CIVREC,CRT,CWH,
        CSA,CSAREC,CSH,CGT,CTREC,PCD,CQTU,PCP,CQTM,
        PCR,CQTF,PCS, CQTUMP,PCGT,SEMPT,SEPC,SPCR
        SEPV,SDEL,SPDEL,HPD(5,2),CAYZ(10),CAYZI(10),
        PERS(5,2)
```

This common block contains the cost elements accumulated for each LRU into a case total. This common block is referenced in Subroutine OPER and the eight output routines called by OPER.

9.
```
COMMON/INPUT/ARA,AYZP,CAD,CALMAN,CALPUB,CALSET,
        CCAL,CCALP,CCALR,CCSP,CCSPP,CCSPR,CDDI,CDEO,
        CDFD,CDID,CDIO,CDIST,CDMAN,CDOE,CDOI,CDPMAN,
        CDPRMN,CDRMAN,CEMAN,CEN,CEND,CERMAN,
        CFTD,CGMAN,CGRMAN,CI,CII,CKIT,CKMD,CKME,CKMI,
        CKMO,CKPD,CKPI,CKPO,CKUD,CKUE,CKUI,CKUO,CLRUPG,
        CMODPG,CMP,CONMAN,CONTCT,CPE,CPI,CPII,CPP,CPUBII,
        CRI,CRII,CRM,CRP,CRU,CSDEP,CSDSU,CSESU,CSGSU,
        CTCPUB,CTRA,CTRCAL,CTRI,CTRII,CTRSPT,CUBEM,
        CUBEP,CUBEU,CUCE,CUP,DAOQL,DD,DDS,DI,DIS,E,ED,
        EDS,EE,EVDM,EVDR,EVDT,EVEM,EVER,EVET,EVIM,EVIR,
        EVIT,EVOM,EVOR,EVOT,FI,FII,FINT,FMD,FMI,FMO,
        FN,FNGF,FNSP,FSA,FTI,FTII,FTM,FTP,FTU,FUD,
        FUE,FUI,FUO,HPM,HPP,HPU,OD,ODS,OTF,P,PMR,PP,
        PPR,PUR,QMM,QMP,QMU,QTD,QTE,QTI,QTMD,QTME,QTMI,
        QTMO,QTO,QTPD,QTPI,QTPO,RDD,REPEAT,RID,ROI,
        SMD,SME,SMF,SMI,SMO,SPE,SPEV,SPEVR,SUD,SUE,SUI,
        SUO,SVE,SVR,SVT,SVV,TALMAN,TATE,TC,TD,TDI,
        TDMAN,TDMW,TDPMI,TDPMII,TDPRI,TDPRII,TDR,TDRMAN,
        TE,TER,TEMAN,TERMAN,TEO,TF,TFR,TGMAN,TGRMAN,TI,
        TID,TIMW,TIO,TIR,TMD,TMDD,TMDR,TMI,TMID,TMIR,
        TMO,TMOD,TMOR,TOE,TOI,TOMW,TONMAN,TRC,TUMD,
        TUMI,TUMO,WD,WDM,WDR,WE,WEM,WER,WI,WIM,WIR,
        WM,WO,WOM,WOR,WP,WTKIT,WU,YAT,YD,YMWO,YP,YR,
        YZ,ZFL,ZI,ZO,H(4),OL(4),OST(4),SL(4),TAT(4),
        TAYZ(10),ZM(4),ZP(3),ZU(4),STAT,DTE,DTO,DTI,
        GA,GB,GC,GD,GE,GF,GG,GH,GI,GJ,GK,GL,GM,GN,GO,
```

```
            GP,GQ,GR,PIN(6),NIN(5),SIN(19)IIN(2),TENMAN,
            CV,PV,CRV,CPUBV,CTRV,FE,ETE WMR,WMT,RF,SAME,
            ETEI,EREI,ILE
```

This common block contains the list of input variables found in NAMELIST/L/ and NAMELIST/LL/. INPUT is referenced by subroutine BLKDAT, BASIC, SUPI, SENSIT, OPER, and the eight subroutines called by OPER.

10. COMMON/SENS/SENSY(266),NRULE(12),NVAR(12),MODE,
            KPASS,NRU,LRU,NPASS

    This common block contains the list of variables that are used to perform sensitivity analyses. SENS is referenced in BLKDAT to initialize the variables and in subroutine SENSIT to print a page header when output page is for a sensitivity analysis.

11. COMMON/SUPIN/C(101),BETI,STI,ISUP,ICALL,AEY,BSAEY,
            AERY,BSAERY,BCAEY,BCAERY,ESU,ESUM,ESUR,ESUY,
            PQTME,CQTME

    This common block contains the summarized LRU data array (C) and other parameters associated with organizational level maintenance. SUPIN is referenced in subroutine SUPI to output data when any of the C, I, J, K maintenance policies are used.

12. DIMENSION G(20),SAVM(22)
    DIMENSION SAVI(229),SAVA(45),SAVG(24),SAVJ(5),
              SAV(339),SAVV(339)
    DIMENSION CUM(75),SUM(35)
    DIMENSION VV(101),TLRU(15)
    DIMENSION UNS(5)
    DIMENSION OSAV(339)
    DIMENSION SUMT(4)
    DIMENSION WPD(5,2)
    DIMENSION DM(278),DR(8)
    DIMENSION SDAM(16)
    DIMENSION PERL(5,2),WPDL(5,2)

    These statements set the dimensions for variables used internal to the main routine (LOGAM1).

13. EQUIVALENCE (CCET,CUM(1))

    This statement equivalences the case total accumulator array CUM to the variable list of Common Block /ZERO/.

14. EQUIVALENCE (GA,G(1),SAVG(1))

    This statement equivalences working arrays G and SAVG to the maintenance policies in COMMON BLOCK /INPUT/. G will occupy the same locations in memory as the 18 maintenance policies plus the first two locations of PIN. SAVG will include the maintenance policies at all locations of PIN(6)).

15. EQUIVALENCE (SAEY,SAVM(1))

This statement equivalences the working array SAVM to SAEY which will later be equivalenced to SIN. SIN is in the variable list of Common Block /INPUT/. SAVM will occupy the same locations in memory as SIN(19), IIN(2), and TENMAN. These three variables occupy consecutive locations in the common block /ZERO/.

16.
```
EQUIVALENCE (H(1),SAVA(1),HE),(H(2),HO),
         (H(3),HI),(H(4),HD)
```
This statement equivalences the working array SAVA and the LRU supply authorization flags HE, HO, HI and HD to the LRU supply authorization array (H) of common block /INPUT/. SAVA is never used in the computations of this routine.

17.
```
EQUIVALENCE (PIN(1),GS), (PIN(2),GT),(PIN(3),EACAL),
         (PIN(4),EACSP)
EQUIVALENCE (PIN(5),ETI),(PIN(6),ETII),(NIN(1),IO),
         (NIN(2),IS)
EQUIVALENCE (NIN(3),JTED),(NIN(4),NA),(NIN(5),NU),
         (SINU(1),SAEY)
EQUIVALENCE (SIN(8),SADRY),(SIN(9),CAEY),
         (SIN(10),CAERY)
EQUIVALENCE (SIN(5),SAIY),(SIN(6),SAIRY),
         (SIN(7),SADY)
EQUIVALENCE (SIN(2),SAERY),(SIN(3),SAOY),
         (SIN(4),SAORY)
EQUIVALENCE (SIN(11),CAOY),(SIN(12),CAORY),
         (SIN(13),CAIY)
EQUIVALENCE (SIN(14),CAIRY),(SIN(15),CADY),
         (SIN(16),CADRY)
EQUIVALENCE (SIN(17),REO),(SIN(18),CTRAD),
         (SIN(19),ARAD)
EQUIVALENCE (IIN(1),IBG),(IIN(2),IOPER)
```
These statements equivalence arrays in common block /INPUT/ to variable names in NAMELIST/L/ and to variables that are computed in LOGAM1. PIN, NIN, and IIN are the arrays that are equivalenced to input variables. SIN, except for 3 elements, is equivalenced to the test and repair manpower demands. The last three elements of SIN are equivalenced to input parameters from NAMELIST/L/.

18.
```
REAL MRO, MRI, MRD, MSO, MSI, MSD
```
This statement sets the module repair and scrap fractions to real numbers (floating point).

B. The DATA statements are used to initialize certain variables to preset values. The DATA statements are briefly discussed here.

1.
```
DATA K001FX/17/
DATA K002FX/18/
```

These statements set the file numbers for storing summarized LRU data. The first summarization will be stored on file 17. After printing the case total, the LRUs will be processed in a new situation and, as each LRU is processed, the information on file 17 is summarized with the new information and written to file 18. The next LRU data will be summarized with the information on file 18 and written to file 17. This alternating procedure between files 17 and 18 continues until all individual LRU data cases are exhausted.

2. DATA K009FX/19/
   This statement assigns file number 19. File 19 is used in the program to store the input data read from cards. The input for each individual LRU case is stored here.

3. DATA K003FX/5/
   DATA K004FX/6/
   These statements assign file units 5 and 6 as the card input file and the printer output files, respectively.

4. DATA NB,ND,INHIB/0,1,0/
   This statement initializes the Program initialization flag NB and the individual LRU print flag INHIB to zero and assigns file number 1 to ND. ND has the same use as K009FX above.

5. DATA SUM/35*0./
   DATA SDAM/16*0./
   DATA WPD/10*0./
   DATA ITR/0/
   These statements initialize accumulator arrays for grand totals, maintenance support costs, and maintenance manhours, respectively. ITR is never used in this program.

C. This section of the program is a series of NAMELIST names and variable lists. Most of the lists are output as a debugging device, others are for inputting LRU data. The NAMELIST names are given below but without the variable lists. Descriptions of the variables can be found in the input and output description lists.

1. NAMELIST/BUG/
   NAMELIST/BUG2/
   NAMELIST/BUG3/
   NAMELIST/BUG4/
   NAMELIST/BUG5/
   NAMELIST/BUG6/
   NAMELIST/BUG7/
   NAMELIST/BUG8/
   NAMELIST/BUG9/
   NAMELIST/BUG10/

These NAMELIST names with their variable lists are used for output purposes only as a debugging device. The variable lists are not included here.

2. NAMELIST/LE/ARA,AYZP,CAD,CALMAN,CALPUB,CALSET,CCAL,
CCALR,CCSP,CCSPP,CCSPR,CDDI,CDEO,CDFD,CDID,
CDIO,CDPMAN,CDPRMN,CDRMAN,CEN,CEND,CFTD,CGMAN,
CGRMAN,CI,CII,CKIT,CKMD,CKMI,CKMO,CKPD,CKPI,
CKPO,CKUD,CKUE,CKUI,CKUO,CLRUPG,CMODPG,
CMP,CONMAN,CONTCT,CPE,CPI,CPII,CPP,CPUBII,
CRI,CRII,CRM,CRP,CRU,CSDEP,CSDSU,CSGSU,
CTCPUB,CTRA,CTRCAL,CTRI,CTRII,CTRSPT,
CUBEM,CUBEP,CUBEU,CUCE,CUP,DAOQL,DD,DDS
DI,DIS,E,ED,EDS,EE,EVDM,EVDR,EVDT,EVIM,
EVIR,EVIT,EVOM,EVOR,EVOT,FI,FII,FINT,
FMD,FMI,FMO,FN,FNGF,FNSP,FSA,FTI,FTII,
FTM,FTP,FTU,FUD,FUI,FUO,HPM,HPP,HPU,OD,
ODS,OTF,P,PMR,PP,PPR,PUR,QMM,QMP,QMU,QT
QTD,QTE,QTI,QTMD,QTMI,QTMO,QTO,QTPD,QTPI,
QTPO,RDD,REPEAT,RID,ROI,SMDA,SMF,SMI,SMO,
SPE,SPEV,SPEVR,SUD,SUI,SUO,SVE,SVR,SVT,
SVV,TALMAN,TATE,TC,TD,TDI,TDMAN,TDMW,
TDPMI,TDPMII,TDPRI,TDPRII,TDR,TDRMAN,
TEO,TF,TFR,TGMAN,TGRMAN,TI,TID,TIMW,
TIO,TIR,TMD,TMDD,TMOR,TMI,TMID,TMIR,
TMO,TMOD,TMOR,TOE,TOI,TOMW,TONMAN,
TRC,TUMD,TUMI,TUMO,WD,WDM,WDR,WI,WIM,
WIR,WM,WO,WOM,WOR,WP,WTKIT,WU,YAT,
YD,YMWO,YP,YR,YZ,ZFL,ZI,ZO,EACAL,
EACSP,ETI,ETII,GA,GB,GD,GE,GF,GG,GH,
GL,GM,GN,GO,GP,GQ,GR,GS,GT,H,IO,INHIB,
IPAGE,IS,ITR,JTED,NA,NB,ND,NU,OL,OST,
SL,TAT,TAYZ,ZM,ZP,ZU,SENSY,STAT,DTO,
DTI,IFLAG,IBG,IOPER,TENMAN,GC,GI,GJ,GK,
TE,TER,WE,WEM,WER,EVEM,EVER,EVET,DTE,
CDIST,TEMAN,FLM,CEMAN,TERMAN,CERMAN,
FUE,SUE,SME,QTME,CKME,CSESU,REO,
CTRAD,ARAD,ILE

This statement contains the lists of input variables that represents and describes an individual LRU data case. Each variable name listed here is described in the input definition section of the document. Most of the data for variables listed here are usually the same from one LRU case to the next.

By using NAMELIST, only the data that changes for the next LRU case needs to be input. The statement that reads this NAMELIST data is described in section F-15 below.

3. NAMELIST/LE/CV,CPV,CRV,CPUBV,CTRV,FE,ETE,WMR,WMT,
RF,ETEI,EREI,ILE

This statement contains a list of NAMELIST input variables

that are optional to the user. To read in this data, ILE=1 must be set when the NAMELIST/L/ data is input. This data is usually a one time input, therefore, ILE=0 can be set either here or in NAMELIST/L/ to prevent reading this input again. These inputs are used to compute work demands and costs for test equipment, specifically Type V test equipment. The description for these variables can be found in the input definition section of this document. The statement that reads this NAMELIST data is described in section F-16 below.

4. NAMELIST/LL/
This NAMELIST is an abbreviated list of the inputs from "LE." Used for output only.

D. The next section of the program initializes variables and reads input data that will be common to all LRU cases.

1. OPERSV=0
OPMNCS=0
These statements initialize the grand total cost and manpower costs used in the post processor section of this program. Subroutine OPER is the driver routine for post processing. OPERSV and OPMNCS are 15th and 16th location of COMMON/POUT/.

2. IPAS=0
This statement initializes a pass counter that counts the number of passes through the cost equations. Used in a debugging output statement.

3. ICALL=0
This statement initializes a flag that denotes when any of the GC, GI, GJ, GK maintenance policies are used by setting the value to 1. Once set in the program ICALL is never used.

4. FLM=0
This statement initializes a multiplication factor for CALSET and CONTCT to denote civilian maintenance labor. The value 1. would be input if civilian maintenance labor was desired.

5. IPAGE=0
This statement initializes the page counter to print with each page of output.

6. ISET=0
IATE=0
These statements initialize the switches that selects the summarized data files K001FX and K002FX.

7.
```
IA7=0
IA8=0
```
These statements initialize the counters that keep track of the number of individual LRU cases that have been stored on the summarized data files K001FX and K002FX.

8.
```
REWIND K001FX
REWIND K002FX
```
These statements assure that the summarized data files are positioned at the beginning of information.

9.
```
ICN=0
```
This statement initializes a counter for the number of distinct LRUs that have been processed in a concept.

10.
```
      READ (K003FX,1000) TEXT,ANALYSIS,DATE,
          COSTIS,AMULT
1000  FORMAT (12A6/12A6/12A6/12A6/3A6/3A6/6A6,
          5X,F10.5)
```
These statements read and set the format for the first set of data cards to be input. Seven cards are input with this format. Four cards for TEXT, one card each for ANALSIS and DATE, and one card containing COSTIS and AMULT.

11.
```
     READ (K003FX,8004)TLRU,IFLAG,NDLRU
8004 FORMAT (15A4, 2I10)
```
These statements read and set the format for the second set of card inputs. Sixty (60) alphanumeric characters are input for TLRU and an integer value each for IFLAG and NDLRU.

E. The next section of the program resets cumulator arrays when reading a completely new set of data.

1.
```
9 ASSIGN 1 tO KAD
```
This statement is the beginning statement for a completely new set of data. The value of one is given to KAD. KAD is checked on in the sensitivity section. When all sensitivity functions are completed KAD will be given the value 9, and return to this section will occur.

2.
```
     DO 8000 I=1,101
8000 C(I)=0.0
     LRU=0
```
These statements reinitialize the summarized LRU accumulator array and the LRU counter to zero.

F. The next section of the program resets inputs and reads new values into memory that are peculiar to the current LRU.

1.
```
1 IF(NB.EQ.0)GO TO 64
```
A return is made to this statement after each individual LRU case is completed. NB is zero on the first pass only. This by-passes the setting of accumulators except on the initial pass through the program.

2.
```
   IF(IS.NE.1)GO TO 52
   DO 53 I=1,339
53 SAV(I)=OSAV(I)
```
These statements recall LRU inputs from a previous read when IS=1.

3.
```
IS=3
```
This statement controls the setting of accumulated cost values and resetting availability, workload, and case total accumulators. IS=3 is set to prevent resetting accumulators until IS is input equal to one at the end of a case (concept).

4.
```
    OPERSV=CGT
    OPMNCS=CCM
    DO 531 I=1,16
531 UVA(I)=SDA(I)
```
These statements store grand total maintenance support costs for output when subroutine OPER is called.

5.
```
64  CONTINUE
    DO 631 I=1,45
631 CUM (I)=0.
```
These statements reset the area of common block/ZERO/ where accumulated values over a concept are additive.

6.
```
    DO 581 I=1,16
581 SDA(I)=0
```
These statements reset the maintenance support cost array before accumulating data for a new grand total. SDA(1) thru SCA(16) is equivalent to F1 thru F16.

7.
```
    DO 641 I=46,65
641 CUM(I)=1.
```
These statements reset the area of common block /ZERO/ where accumulated values over a concept are multiplicative.

8.
```
52 IF(IS.EQ.2)GO TO 50
   DO 51 I=1,20
51 G(I)=0.
50 CONTINUE
```
These statements will reset the maintenance policies for the

next LRU unless IS=2 is input.

9. 65 CONTINUE
This statement is the end of the logic for initializing and resetting totals and accumulator arrays.

10.
```
     READ (K003FX,1101) UNITS
     IF(EOF(K003FX))3,10
1101 FORMAT (5A4)
  10 CONTINUE
```
These statements read the 9th input card and checks for an end of file on the card input device. UNITS is used as an output descriptor of the LRU class and number for the next LRU to be processed. If an end of file is encountered instead of UNITS data, a transfer is made to statement 3 for a program stop. Otherwise, transfer is to statement 10 for additional inputs.

11.
```
     READ (K003FX,1001) REMARK
     IF (EOF(K003FX))3,13
1001 FORMAT(12A6)
     13 REPEAT=1.
```
These statements input the 10th card of data containing 72 characters of information. REMARK is used to describe the next LRU scheduled for input. If an end-of-file is encountered instead of REMARK then transfer is made to statement 3 for a program stop. Otherwise, transfer is made to statement 13 for continuation of LRU inputs. REPEAT is the number of identical LRUs in a material system and must always be at least one.

12.
```
QTE=0.
QTO=0.
QTI=0.
QTD=0.
```
These statements initialize the initial provisions for LRUs at the Organizational, Direct Support, General Support, and Depot supply facilities, respectively.

13.
```
QTME=0.
QTMO=0.
QTMI=0.
QTMD=0.
```
These statements initialize the initial provisions for modules at the Organizational, Direct Support, General Support and Depot supply facilities, respectively.

14.
```
QTPO=0.
QTPI=0.
QTPD=0.
```
These statements initialize the initial provisions for parts at the Direct Support, General Support and Depot supply

facilities, respectively.

15. READ (K003FX,L)
This statement reads LRU data from cards through the NAMELIST/L/ input statement.

16. IF(ILE.EQ.1)READ (K003FX,LE)
This statement reads Type V test equipment data using NAMELIST/LE/. This read can be turned on by inputting ILE=1 in NAMELIST/L/ and turned off (ILE=0) in either /L/ or /LE/ NAMELIST.

17. IF(IBG.EQ.1)WRITE(6,L)
This statement prints the input list from NAMELIST/L/. Input IBG=1 in NAMELIST/L/ to get this output.

18. IF((ILE.EQ.1).AND.(IBG.EQ.1)) WRITE(6,LE)
This statement prints the additional LRU inputs from NAMELIST/LE/.

19. IF(AB(GC+GI+GJ+GK).NE.0.)ICALL=1
This statement sets the flag ICALL=1 when a maintenance concept is input for LRU repair at the equipment level.

20.
```
   IF(NB .NE. 0)GO TO 54
   NB=1
   DO 55 I=1339
55 OSAV(I)=SAV(I)
54 CONTINUE
```
These statements save the data of COMMON/INPUT/ (as modified by the last NAMELIST/L/ input) for possible recall on a later pass. SAV and INPUT are equivalent. NB is set equal to 1 to prevent accidentally resetting OSAV.

21. IF(IOPER.EQ.1.AND.NU.EQ.-4)CALL OPER
IF(NU.EQ.-4)STOP
When an analysis is complete NU=-4 is input through NAMELIST/L/ to terminate program execution. IF IOPER is input as 1 then subroutine OPER is called before terminating the program to compute and print operating and support costs.

22. IF(SENSY(1).NE.0.)REWIND KOO9FX
IF(SENSY(1).NE. 0.)GO TO 11
These statements rewind a scratch tape and transfer control to the sensitivity section of the program SENSY data is read with the NAMELIST/L/ input list. SENSY (1) is the number of variables in the input list that will undergo modifications for a sensitivity run. SENSY data should be input only after all LRUs in a concept have been processed.

23. IF(QTE.GE..5)H(1)=1.

```
IF(QTD.GE..5)H(2)=1.
IF(QTI.GE..5)H(3)=1.
IF(QTD.GE..5)H(4)=1.
IF(H(4).LT..5)HPU=0.
```

These statements check the NAMELIST/L/ input values for predetermined quantities of LRU stock at the Equipment, Direct, General and Depot facilities. If quantities are input (>.5) at a facility, then the stock authorization flag is stored in the appropriate H location. Discretionary procurement holding time for LRUs (HPU) is set to zero when there is not LRU stock at the Depot.

24.
```
    DO 150 I = 1,4
    IF (H(I).GE.0.5)GO TO 151
150 CONTINUE
    H(4)=1.
```

These statements check the stock authorization array (H) to determine if stock is authorized for at least one stock level. If a stock level was not authorized than the program sets authorization at the Depot.

25.
```
151 GTOT=0.
    DO 152 I = 1,20
152 GTOT=GTOT+G(I)
```

These statements sum the maintenance policy fractions (G). The sum of maintenance policies input must sum to 1. GTOT will be used to test for correct maintenance input policies.

26.
```
IF (GTOT.GT..9999.AND.GTOT.LT.1.0001)GO TO 157
```

This statement checks for maintenance policies that do not sum to 1 (100% of maintenance). If the total is 1.0 then logic is transferred to statement 157, otherwise the following statements are executed.

27.
```
    WRITE (K004FX,153)
    DO 154 I = 1,20
    BG=G(I)
    G(I)=BG*(1.0/D(GTOT))
154 WRITE (K004FX,155)I,BG,G(I)
    WRITE (K004FX,156)GTOT
```

These statements are executed when the G policies do not sum to 1.0. A message informing the user of the error is printed and the input policies are prorated as a function of the input total (GTOT). The modified G policies that now sum to 1.0 are printed along with policies input (BG). The total (GTOT) of the G policies that are input is also printed.

28.
```
157 CONTINUE
```

This is the statement transferred to when the maintenance (G) policies sum to 1.0.

29.
```
IF(LRU.EQ.0)REWIND ND
LRU=LRU+1
```
These statements rewind the individual LRU data file on the first pass of a LRU concept. Each LRU data record in a concept will later be written to file ND. LRU is then incremented for each NAMELIST/L/ data record input. LRU is reset to zero after a sensitivity run.

30.
```
     IF(IFLAG.GT.0)GO TO 8472
     ICN=ICN+1
     IF(ICN.GT.NDLRU)ICN=1
     DO 8372 LVZ=1,5
8372 UNS (LVZ)=UNITIS (LVZ)
```
These statements control the setting of a case total summation counter and LRU unit descriptions when IFLAG=0. NDLRU is the number of LRU cases in a summation set. When ICN becomes greater than NDLRU, ICN is then reset (ICN=1) so that the next NDLRU set of LRUs will be summed with the previous set (NDLRU) of LRUs.

31.
```
8472 CONTINUE
```
This statement is the transfer point when LRU summations are not requested (IFLAG=1).

32.
```
     IF(IO.EQ.2)WRITE(6.L)
     IF(ILE.EQ.1).AND.(IO.EQ.2))WRITE(6,LE)
     IF(IO.EQ.2)WRITE(6,1002) DATE,UNITIS,ANLYIS
1002 FORMAT(5X,7HDATE-,3A6,5X7HUNIT-,5A4,
           5X11HANALYSIS-3A6)
     IF(IO.EQ.1)WRITE(6,LL)
```
These statements print the NAMELIST inputs and LRU descriptions when the flag IO=2 is set in NAMELIST/L/ input.

33.
```
WRITE(ND)SAVV,UNITS,REMARK
WRITE(K009FX)SAVV,UNITS,REMARK
```
These statements write the NAMELIST inputs (SAVV) and LRU descriptions to data files ND and K009FX. ND is the file read in the sensitivity section of this program to retrieve the data for each individual LRU case. The ND file is also used to print formatted inputs. K009FX is a scratch file and is used only to retrieve the LRU unit descriptions (UNITS) for printing with summarized LRU outputs.

34.
```
IF(IO.LT.3)GO TO 12
```
This statement is used to print the NAMELIST inputs in a formatted form by inputting IO=3. The formatted output write statements follow this statement in the program listing, but are not included in this program description.

G. This next section of the program reads the input data file ND and prints the data in a formatted form. Since the names in the output lists are the same as in the NAMELIST list, an explanation of this section of code by line item will be omitted.

1. GO TO 12
   This statement skips to the section of the program that begins the logistics computation.

H. The next section of the program is the sensitivity section. After all LRUs in a concept are processed, inputs to the SENSY array can alter selected parameter values. Modifying the original LRU values with the SENSY inputs is performed in the sensitivity section.

1. 11 CONTINUE
   This is the transfer statement after a sensitivity request is made. Transfer to this statement is described in line item F-22 above.

2. IF(LRU.EQ.0)LRU=NRU
   This statement is required only if sensitivity cases are stacked (more than one case). When a sensitivity case is completed, program transfer is made to statement number 1 where LRU is reset to zero before reading the next data case. If another sensitivity case is input then NRU will still have the value for the number of LRUs processed in the last sensitivity case; whereby LRU can be properly reset.

3. IF(LRU.EQ.0) GO TO 91
   This statement transfers logic to a diagnostic statement when LRU=0. The only time this should happen is when SENSY(1) is input with the first LRU data case and has a value greater than zero.

4. MODE=SENSY(1)+.1
   This statement sets the number of variables in NAMELIST/L/ that are to be modified. The ".1" is to solve round-off problems when converting from floating point to integer.

5. NPASS=SENSY(2)+.1
   This statement sets the number of variations for a selected set of variables (NVAR) to be modified. Essentially this is the number of passes that will be made through the LRU data base of file ND. Each pass modifies LRU data values according to the rules (NRULE) which are explained later.

6. IF ((MODE.GT.12).OR.((2+(MODE*(NPASS+2))).GT.266))
   GO TO 19

This statement checks for the number of values that can be input to the SENSY array without overflowing the dimensioned value (266) of the array. If overflow occurs transfer is made to a diagnostic statement (19).

7.
```
  DO 6 I=1,MODE
  NRULE(I)=SENSY(I+2)+.1
  MODEI2=MODE+2+I
6 NVAR(I)=SENSY(MODEI2)+.1
```
These statements set the rules (NRULE) for variable modification and the variable's position (NVAR) in the labeled common block/INPUT/. The modification rules are explained later in line item H-21.

8. IF (INHIB.EQ.1) CALL PAGE
This statement positions the printer paper to the top of a page and prints header information when individual LRU printout is inhibited.

9. KPASS=0
This statement initializes the pass counter.

10. 7 KPASS=KPASS+1
This statement counts the number of passes through an LRU data set. Each pass is a variation to the NVAR set of inputs. When all the LRUs in a data set have been processed, return is made to this statement unless all input variations have been exhausted (KPASS=NPASS).

11. REWIND ND
This statement rewinds the LRU data file.

12. NRU=0
This statement initializes a counter for the number of LRUs analysed in a sensitivity pass.

13. 8 NRU=NRU+1
This statement counts the number of LRUs that have been modified and processed. Return will be made to this statement after each LRU is processed until NRU=LRU, where LRU is the number of individual LRU data cases analysed in the baseline case.

14.
```
    IF (IS.EQ.2)GO TO 56
    DO 57 I=1,20
 59 SAV(I)=OSAV(I)
    IS=3
    DO 611 I=1,45
611 CUM(I)=0.
    DO 582 I=1,16
582 SDA (I)=0.
```

```
    DO 621 I=46,75
621 CUM (I)=1.
 58 CONTINUE
    IF(IS.EQ.2)GO TO 56
    DO 57 I=1,20
 57 G(I)=0.
 56 CONTINUE
```

These statements reset the maintenance policies (G), input data case (SAV), and cumulative data arrays (CUM, SDA). Their use and description is the same as described in line items F-1. through F-8.

15.
```
IF(QTE.GE..5)H(1)=1.
IF(QTO.GE..5)H(2)=1.
IF(QTI.GE..5)H(3)=1.
IF(QTD.GE..5)H(4)=1.
IF(H(4). LT..5) HPU=0.
```
These statements set initial provision flags when LRU stock quantities are input at Equipment, Direct Support, General Support, and Depot, respectively. The description is the same as described in line item F-23.

16. IF(NRU.LT.LRU) ASSIGN 8 TO KAD
This statement assigns the value 8 to the logic transfer variable KAD until all LRUs of the baseline case have been processed. KAD=8 sends control to statement number 8 after a sensitivity pass through the baseline case is completed. Statement number 8 is described in line item H-12.

17. IF (NRU.EQ.LRU) ASSIGN 7 TO KAD
This statement assigns the value 7 to the logic transfer variable KAD when the last LRU in a data set is processed. KAD=7 sends control to statement number 7 for the next sensitivity pass of the LRUs. Statement number 7 is described in line item H-9.

18. IF (NRU.EQ.LRU.AND.KPASS.EQ.NPASS) ASSIGN 9 TO KAD
This statement assigns the value 9 to the logic transfer variable KAD when all sensitivity passes for all LRUs have been exhausted. KAD=9 sends control to statement number 9 to reinitialize program variables and input a new data set. Statement number 9 is described in E-1.

19. READ (ND) SAVV, UNITIS, REMARK
This statement reads the NAMELIST inputs (SAVV) for the current LRU case to be modified for a sensitivity run. The LRU units and remarks information are also read from the file.

20.
```
   I=0
80 I=I+1
```
These statements initialize and set the counter for

selecting the modification rule and the position of the variable (within the /INPUT/ list) to be modified.

21. KRULE=NRULE(I)
IF(KRULE.LT.1.OR.KRULE.GT.5)GO TO 300
These statements set the rule number for modifying a variable and checks if it is an acceptable number. There are only 5 possible rules; replacement, addition, subtraction, multiplication and division. If any other rule number is input a transfer to a diagnostic statement is made.

22. M=NVAR(I)
This statement selects the position within the COMMON/INPUT/ variable list for the Ith variable to be modified.

23. INDEX=I+2+((KPASS+1)*MODE)
This statement computes the position within the SENSY array that contains the value used to modify the LRU data variable represented by location M.

24. IF(M.LT.1.OR.M.GT.249)GO TO 200
This statement transfers to a dignostic statement if the variables position is not within the first 249 positions of COMMON/INPUT/.

25. VALUE=SENSY(INDEX)
This statement stores the modification value for the Ith MODE and RULE.

26. GO TO (81,82,83,84,85),KRULE
This statement sends control to the program logic associated with the rule currently in effect (KRULE)

27.
```
81 SAV(M)=VALUE
   GO TO 86
82 SAV(M)=SAV(M)+VALUE
   GO TO 86
83 SAV(M)=SAV(M)-VALUE
   GO TO 86
84 SAV(M)=SAV(M)*VALUE
   GO TO 86
85 IF(VALUE.EQ.0)GO TO 100
   SAV(M)=SAV(M)/VALUE
```
These statements modify the inputs (SAV) for an individual LRU baseline case according to the value of KRULE. The rule for modification is; KRULE=1, replace the old value with new VALUE; KRULE =2, add old value to VALUE; KRULE=3, subtract old value and VALUE; KRULE=4, multiply old value by VALUE; KRULE=5, divide old value by VALUE. If VALUE=0 and KRULE=5 a transfer is made to a diagnostic statement.

28.
```
86 IF(I.LT. MODE)GO TO 80
```
This statement checks to determine if all the sensitivity modes have had their values modified.

29.
```
IF(INHIB.EQ.1)CALL SENSIT
```
This statement calls subroutine SENSIT to print a description of the sensitivity run when the individual LRU print flag is turned off.

30.
```
IF(NRU.LT.LRU)GO TO 12
IF(KPASS.LT.NPASS)GO TO 12
```
These statements transfer control to the logistics computation section of this program until all sensitivity passes for all individual LRU cases are processed.

31.
```
   DO 90 J=1,INDEX
90 SENSY(J)=0.
   GO TO 12
```
These statements reinitialize the SENSY array when the last pass of the last LRU is reached. Transfer of control is then made to the logistics section to compute the last sensitivity pass.

32.
```
100 WRITE (K004FX,102)INDEX
102 FORMAT(1X,40HATTEMPTED DIVISION BY
         ZERO, ERROR INDEX,=I5)
    GO TO 86
```
These statements print a diagnostic message when the value input to SENSY for a division rule (KRULE=5) is zero and then sends control to check for the presence of another variable to modify.

33.
```
200 WRITE (K004FX,202) M
202 FORMAT (1X,26HILLEGAL VARIABLE ADDRESSED,I4)
    GO TO 86
```
These statements print a diagnostic message when the variable selected for modification is not one of the permitted ones; i.e., variables that occupy locations 1 through 249 of COMMON/INPUT/. Control is then sent to statement 86 to test for the presence of another variable to modify.

34.
```
300 WRITE (K004FX,302) KRULE
302 FORMAT (1X,19HILLEGAL RULE KRULE=I5)
    GO TO 86
```
These statements print a diagnostic message when a modification rule is not one of the permitted ones; i.e., KRULE is not 1 through 5. Control is sent to statement 86 to test for the presence of another variable to modify.

I. The next section of programming begins the basic logistics flow on a per LRU per hour basis. This section contains equations and operations in which the demands that each LRU type makes on the support system and the costs consequently incurred are calculated. With the exception of line item 2 below the computations in this section are for only one equipment installation.

1. 12 CONTINUE
This statement is the starting control point for logistics computations for all LRU types.

2. EDEE=ED*EE
This statement computes the number of LRUs of the same type for all equipment installations.

3. F=E*EE
This statement computes the number of failures per hour per installation.

4. FNG=FNGF*F
This statement computes the number of false "no go's" per hour per installation.

5. A=YAT*EE/(8766.*OTF)
This statement computes the hourly attrition rate of LRUs per installation based on operating hours of the materiel system.

6. FMWO=YMWO/8766
This statement computes the number of Modification Work Orders (MWOs) per hour per LRU.

7. YPF=(8766.*YP)-(168.*FTU)
This statement computes the length of the acquisition phase in hours, taking into account the factory start-up time (FTU) for LRUs.

8. IF(YPF.LT.168.*FTU) YPF=168.*FTU
This statement will set the acquisition phase time equal to the factory start-up time for LRUs if input acquisition time is less than the factory start-up time.

9. If (YPF.LE.0) YPF=2000
This statement sets the acquisition phase to 2000 hours when neither the factory start-up time for LRUs nor an acquisition time is input.

10. IF(IBG.EQ.1)WRITE(6,BUG)

This statement is used to debug a set of inputs. NAMELIST/BUG/ is printed when IBG=1 is input.

11. SUEC=1.-SUE
SUOC=1.-SUO
SUIC=1.-SUI
SUDC=1.-SUD
These statements compute the fraction of LRUs not scrapped at the Equipment, Direct Support, General Support and Depot levels, respectively.

12. SMEC=1.-SME
SMOC=1.-SMO
SMIC=1.-SMI
SMDC=1.-SMD
These statements compute the fraction of modules not scrapped at the Equipment, Direct Support, General Support and Depot levels, respectively.

13. FUEC=1.-FUE
FUOC=1.-FUO
FUIC=1.-FUI
FUDC=1.-FUD
These statements compute the fraction of LRUs not repaired at the Equipment, Direct Support, General Support and Depot levels, respectively.

14. FMOC=1.-FMO
FMIC=1.-FMI
FMDC=1.-FMD
These statements compute the fraction of modules not repaired at Direct Support, General Support and Depot levels, respectively. (Modules are not repaired at equipment level).

15. REOT=24.*(REO+TOE)
ROIT=24.*(ROI+TIO)
RIDT=24.*(RID+TDI)
These statements compute the hours of supply for condemned modules and parts at the Equipment, Direct Support and General Support levels, respectively.

16. TEOT=TEO+TOE
This statement computes the two-way pipe time in hours for the Equipment/Direct Support pipe.

17. TOIT=24.*(TOI+TIO)
This statement computes the hours of supply at Direct Support for LRUs (repaired or condemned) and modules that will be repaired.

18. TIDT=24.*(TID+TDI)

This statement computes the hours of supply at General Support for LRUs and for repaired modules.

19. SHTEO=CDEO+CD0E
SHTOI=CD0I+CDI0
SHTID=CDID+CDDI
These statements compute the round trip shipping costs in dollars per pound between Equipment/Direct Support, Direct Support/General Support, and General Support/Depot, respectively.

20. USE=SUE+SUEC*FUEC
URE=SUEC*FUE
ULE=SUEC*FUEC
USO=SUO+SUOC*FUOC
URO=SUOC*FUO
ULO=SUOC*FUOC
USI=SUI+SUIC*FUIC
URI=SUIC*FUI
ULI=SUOC*FUIC
These statements compute the fraction of LRUs scrapped, the fraction of LRUs repaired and the fraction of LRUs in limbo at the Equipment, Direct Support and General Support levels, respectively.

21. USD=SUD+SUDC*FUDC
URD=SUDC*FUD
These statements compute the fraction of LRUs scrapped and repaired at Depot, respectively.

22. MS0=SMO+SMOC*FMOC
MRO=SMOC*FM0
MSI=SMI+SMIC*FMIC
MRI=SMIC*FMI
MSD=SMD+SMDC*FMDC
MRD=SMDC*FMD
These statements compute the fraction of modules scrapped and the fraction of modules repaired at Direct Support, General Support and Depot, respectively.

23. IF(IBG.EQ.1)WRITE (6,BUG)
This statement is used to print intermediate calculations for debugging purposes.

24. TU=A+F+FNG
This statement computes LRU removals per hour by summing attrition, failures, and false removals.

25. DPT=2.-DAOQL
This statement computes a workload factor for Depot repairs as a result of faulty repairs being returned.

26. DSUE1=(F+FNG)*(GB+GD+GF+GL+GM+GN+GO+GP)+
(F*DPT+FNG)*(GH+GQ)
GSUE1=(F+FNG)*(GE+GR+GS)
DUE1 = (F*DPT+FNG)*(GG+GT)
These statements compute an hourly flow of LRUs from the Organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to evacuate failures and false "no go's" to a higher maintenance level.

27. DSUE2=F*ULE*GI
GSUE2=F*ULE*GH
DUE2 =F*ULE*GK
These statements compute the hourly flow rate for "in limbo" LRUs from the organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to repair at ED, but were not repaired because of the FUE fraction; therefore, resulting in an evacutation to a higher maintenance level.

28. FUTE=F*(GC+GI+GJ+GK)
This statement computes the hourly flow of LRUs through the test facility at the organizational level.

29. SUE1=(F+FNG)*GA
SUE2=F*ULE*GC
SUE3=SUE*FUTE
These statements compute hourly scrap rates for LRUs at the Organizational level. These scrap rates are computed when there is not a repair maintenance policy, when "in limbo" LRUs do not have a higher level of maintenance, and for those LRUs that are scrapped after going through the test facility, respectively.

30. FNG=FNG*(GC+GI+GJ+GK)
This statement computes the hourly flow of false "no go's" which are screened at Organizational level, found serviceable and returned to stock.

31 TUFTE=URE*FUTE
This statement computes the hourly flow of LRUs through the test facility at the Organizational level that are deemed repairable.

32. TUFEC=TUFTE+FGE+SUE3+SUE2+DUE2+GSUE2+DSUE2
This statement computes the hourly flow of LRUs that go through the checkout facility at the Organizational level. These are all the flows at ED that will be charged TC.

33. GSUO1=F*SUOC*(GF+GO+GP)
DUO1=F*SUOC*(GH+GQ)*DPT

These statements compute the hourly flow of LRUs that are evacuated from the Direct Support level to General Support and Depot, respectively. These evacuations are for LRUs with maintenance policies to screen false "no go's" at Direct Support and send the non-scrapped fraction of failures to a higher echelon of maintenance.

34. GSUD2=F*ULO*GM
DUO2 =F*ULO*GN
These statements compute the hourly flow rate for "in limbo" LRUs from Direct Support to General Support and Depot, respectively. These flow rates are for the units that have a maintenance policy to repair at OD but were not repaired because of the FUO fraction; therefore, resulting in an evacuation to a higher maintenance level.

35. SUO1=F*SUO*(GF+GO+GP)
SUO2=F*SUO*(GH+GQ)*DPT
These statements compute the hourly scrap rate for LRUs at the Direct Support level. The scrap rates here are for those maintenance policies that screen false "no go's" at Direct Support and evacuate LRUs for repair to General Support and Depot, respectively.

36. SUO3=F*GB
This statement computes the hourly scrap rate for LRUs using the maintenance policy to detect false "no go's" at Direct Support and discard all failures.

37. SUO4=F*ULE*ULO*(GD+GL)
This statement computes the hourly scrap rate at Direct Support for that fraction of "in limbo" LRUs evacuated from ED and are still having an "in limbo" status at Direct Support.

38. FUTF=F*(GD+GL+GM+GN+ULE*GI)
This statement computes the hourly flow of LRUs through the test facility at Direct Support.

39. SUO5=SUO*FUTF
This statement computes an hourly scrap rate for LRUs at Direct Support. The scrap rate computed here is for those LRUs that flow through the test facility and are declared not repairable.

40. FGO=FNG*(GB+GD+GF+GH+GL+GM+GN+GO+GP+GQ)
This statement computes the hourly flow of false "no go's" screened at Direct Support, found serviceable, and returned to stock.

41. TUFTO=URO*FUTF
This statement computes the hourly flow of LRUs through the

test facility at Direct Support that was deemed repairable.

42. TUFOC=TUFTO+FGO+SUO5+SUO4+SUO3+SUO2+SUO1+DUO2+GSUO2
+DUO1+GSUO1
This statement computes the hourly flow of LRUs through the checkout facility at Direct Support. These are all the flows that will be charged TC at OD.

43. DUI=F*ULI*SUOC*GP+F*ULI*GS
This statement computes the hourly flow rate for "in limbo" LRUs from General Support to Depot. This flow is for units that are neither repaired nor scrapped at GS.

44. SUI1=F*ULI*(GE+GR+ULE*GJ+ULO*GM+SUOC*(GO+GF))
This statement computes the hourly scrap rate of LRUs at General Support for "in limbo" units evacuated from ED and DS and are still "in limbo" status at General Support. These LRUs are scrapped because there is not a higher maintenance level for evacuation.

45. FUTI=F*(GE+GR+GS+ULE*GJ+ULO*GM+SUOC*(GO+GF+GP))
This statement computes the hourly flow of LRUs through the test facility at General Support.

46. SUI2=SUI*FUTI
This statement computes the hourly scrap rate for LRUs at General Support. The scrap rate here is for these LRUs that flow through the test facility and are declared not repairable.

47. FGI=FNG*(GE+GR+GS)
This statement computes the hourly flow of false "no go's" screened at General Support, found servicable, and returned to stock.

48. TUFTI=URI*FUTI
This statement computes the hourly flow of LRUs through the test facility at General Support that are deemed repairable.

49. TUFIC=TUFTI+FGI+SUI2+SUI1+DUI
This statement computes the hourly flow of LRUs through the checkout facility at General Support. These are all the flows at DI that will be charged TC.

50. FUTD=F*((GG+GT)*DPT+ULE*GK+ULO+GN+ULI*GS+SUOC*
DPT*(GH+GQ)+ULI*SUOC*GP)
This statement computes the hourly flow of LRUs through the test facility at Depot.

51. SUD=USD*FUTD
This statement computes the hourly scrap rate for LRUs at the Depot. The scrap rate computed here is for those LRUs

that flow through the test facility at Depot and are declared not repairable.

52. FGD=FNG*(GG+GT)*DPT
This statement computes the hourly flow of false "no go's" screened at Depot, found servicable, and returned to stock.

53. TUFTD=URD*FUTD
This statement computes the hourly flow of LRUs through the Depot test facility that are deemed repairable.

54. TUFDC=TUFTD+FGD+SUD
This statement computes the hourly flow of LRUs through the checkout facility at Depot. These are all the flows at DD that will be charged TC.

55. DSME=F*URE*SMEC*GI
GSME=F*URE*SMEC*GJ
DME =F*URE*SMEC*GK*DPT
These statements compute hourly flow of modules from the Organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to evacuate failures to a higher maintenance level.

56. TMFEO=DSME+GSME+DME
This statement computes the total flow of modules evacuated from the Organization level to higher echelons of maintenance.

57. SME1=F*SM1*URE*(GI+GJ+GK)
This statement computes a scrap rate at the organizational level for those modules tested at ED and are declared not repairable.

58. SME2=F*URE*GC
This statement computes a scrap rate for modules at the Organizational level when the Maintenance policy is to repair an LRU and discard the module.

59. TSME=SME1+SME2
This statement computes the total hourly scrap rate at the Organizational level.

60. GSMO=F*URO*SMOC*GM
DMO =F*URO*SMOC+GN*DPT
These statements compute the hourly module flow from Direct Support to General Support and Depot, respectively, for maintenance policies that evacuate failures to a higher maintenance level.

61. SMO1=F*URO*(MSO*GL+ULE*MSO*GI)+F*URE*SMEC*MSO*GI

SMO2=F*URO*SMO*GM
SMO3=F*URO*GD
SMO4=F*URO*SMO*GN
These statements compute the hourly module scrap rates at Direct Support. The scrap rates computed here are for maintenance policies that repair at DS, evacuatate modules for repair at GS, discard modules at DS, and evacuate modules for repair at Depot, respectively.

62. TSMO=SMO1+SMO2+SMO3+SMO4
This statement computes the total module scrap rate for modules at Direct Support.

63. TMFO=F*MRO*(URO*GL+URO*ULE*GI+URE*SMEC*GI)
This statement computes the hourly flow of modules through the repair facility at Direct Support.

64. TMFOI=GSME+DME+GSMO+DMO
This statement adds the hourly flow of modules from Direct Support to higher maintenance echelons to the flow originating at the organizational level.

65. DMI=F*URI*(SMIC*GS+SUOC*SMIC*GP)*DPT
This statement computes the hourly flow of modules from General Support when the maintenance policies are to repair at the Depot level.

66. SMI1=F*MSI*(URE*SMEC*GJ+URO*SMOC*GM+URI*
(GR+SUOC*GO+ULE*GJ+ULO*GM))
SMI2=F*URI*(GE+SUOC*GF)
SMI3=F*URI*(SMI*GS+SUOC*SMI*GP)
These statements compute the hourly scrap rates for modules at General Support. The rates computed here are for maintenance policies that repair at GS, discard at GS, and evacuate for repair at Depot, respectively.

67. TSMI=SMI1+SMI2+SMI3
This statement computes the total module scrap rate at General Support.

68. TMFI=F*MRI*(URI*(ULO*GM+ULE*GJ+GR*SUOC+GO)+
URE*SMEC*GJ)
This statement computes the hourlv flow of modules through the repair facility at General Support.

69. TMFID=DME+DMO+DMI
This statement computes the total module flow from the lower echelons to the repair facility at Depot.

70. FSMD1=F*URD*(DPT*GT+ULE*GK+ULO*GN+ULI*GS+
SUOC*DPT*GQ+ULI*SUOC*GP)
FSMD2=F*(URE*SMEC*GK+URO*SMOC*GN+URI*SMIC*GS+

URI*SUOC*SMIC*GP)*DPT

These statements compute the hourly flow of modules through the test facility at Depot. FSMD1 is for the maintenance policies that repairs modules at Depot with LRU repair at Depot and the "in limbo" units from lower echelons. FSMD2 is for maintenance policies where LRUs are repaired at lower echelons and the modules are evacuated to the Depot.

71. SMD1=MSD*(FSMD1+FSMD2)
SMD2=F*URD*(DPT*GG+SUOC*DPT*GH)
These statements compute the hourly scrap rate for modules at Depot. SMD1 is the scrap rate for the modules that go through the test facility. SMD2 includes the scrap for the "discard" maintenance policy and the scrap after screening for false "no go's".

72. TSMD=SMD1+SMD2
This statement computes the total scrap rate for modules at the Depot level.

73. TMFD=MRD*(FSMD1+FSMD2)
This statement computes the hourly flow of modules through the Depot repair facility.

74. TSLF=A+SUE1+SUE2+SUE3+SUO1+SUO2+SUO3+SUO4+SUO5+SUI1+SUI2
This statement sums the LRU hourly scrap rates of all Field level facilities.

75. TSLD=SUD
This statement sets the LRU hourly scrap rate at the Depot level.

76. TSU=TSLF+TSLD
This statement computes the total hourly scrap rate for LRUs.

77. TSPO=TMFO
TSPI=TMFI
TSPD=TMFD
These statements set the parts demand flow at Direct Support, General Support and Depot levels, respectively. Since each repaired module requires a new part, the parts demand and module demand are the same.

78. IF (IBG.EQ.1) WRITE(6,BUG)
This statement prints NAMELIST/BUG/ when debugging of a data case is required.

79. TMWO=FMWO*ZO
TMWI=FMWO*ZI
These statements compute the number of modification work orders per hour per LRU at Direct Support and General

Support, respectively.

80. HF=(24.*HPU)+(168.*FTU)
HM=(24.*HPM)+(168.*FTM)
HP=(24.*HPP)+(168.*FTP)
These statements compute the hours to start procurement of LRUs, modules and parts, respectively.

81. CALL BASIC
(CDFD,CDFD,CDFD,CDFD,CDFD,CDFD,0.,CDOE,CDIO,
CDDI,0.,0.,0.,0., SHTEO,SHTOI,SHTID)
This statement calls subroutine BASIC to compute the cost of shipping and handling of LRUs, modules and parts. The arguments for BASIC are the shipping rates in dollars per item per trip between Factory and Depot, Depot and General Support, etc.

82. SHU=QYF+(FMWO*(SHTEO+((1.-ZO)*SHTOI)
+((1.ZO-ZI)*(SHTID*DPT))))
SHM=QMO+QMI+QMD+QFME+QFMO+QMFI+QFMD+QME
SHP=QPO+QPI+QPD
These statements compute the shipping and handling cost factors in dollars per pound of all pipeline flow for LRUs, modules and parts, respectively.

83. SHMF=SHM-QMD-QFMD
SHPF=SHP-QPD
These statements compute shipping and handling cost factors with the exclusion of the Depot/factory pipe for modules and parts, respectively.

84. SHKIT=FMWO*(CDFD+(CDDI*(ZO+ZI))+(CDIO*ZO))
This statement computes the total cost factor for modification Work Orders (MWOs).

85. SHKITF=FMWO*CDIO*ZO
SHKITD=SHKIT-SHKITF
These statements compute the MWO shipping cost factors to Field and Depot, respectively.

86. SHUF=QYF+(FMWO*(SHTEO+(1.-ZO)*SHTOI)))
SHUD=SHU-SHUF
These statements compute LRU shipping cost factors to the Field and Depot, respectively.

87. CALL BASIC
(168.*FTM,168.*FTP,168.*FTU,HF,HM,HP,
24.*RDD,REOT,ROIT,RIDT,TUMD,
TUMI,TUMO,TRC,TEOT,TOIT,TIDT)
This statement calls subroutine BASIC to compute the quantity of LRUs, modules and parts "down" or tied up (in the pipelines) per hour per LRU location.

88. TMWD=FMWO*(1.-ZO-ZI)
This statement computes the frequency of modification work orders at the Depot level.

89. QYZ=QYE*H(1)+(QYO*H(2)*(1.-H(1)))+)QYI*H(3)
(1.-H(1))(1.-H(2)))+(QYD*H(4)*(1.-H(1)
(1.-H(2)(1.-H(3)))+(1.-H(1))*(1.-H)(2))
(1-H(3))(1.-H(4)*QYF
This statement computes the quantity of "down" LRUs depending on the authorized LRU supply locations (H). For example if none of the locations (Equipment, Direct, General, Depot) are authorized supply points (H(1)=0.) then the quantity is QYF, the quantity determined in BASIC by including the Depot/Factory pipe.

90. AYZ=EE/(EE+OTF*QYZ)
This statement computes the inherent availability of the system at an LRU installation.

91. AYZIS=SPOL(AYZ,FN,EE)
This statement calls subroutine SPOL to modify inherent availability (AYZ) to account for allowable failures (FN).

92.
```
    DO 933 I=1,NA
933 IF(TAYZ(I).EQ.1)CAYZI(I)=CAYZI(I)*
    (AYZIS**REPEAT)
```
This DO loop tallies inherent availabilities for subsystems of LRUs. The system availability is tallied in the first location and each subsystem's tally will be recorded in each subsequent location. The LRU subsystem structure is determined by the location of the TAYZ inputs within the LRU data deck (NAMELIST/L/). CAYZI is used to print out subsystems availabilities on the case total output page.

93. SAVE=ED*OTF*AYZ
SAVP=SAVE/D(P)
SAVPP=SAVE/D(PP)
These statements compute the "real time" operating factors for all equipment installations as a function of inherent availability for LRUs, Modules and Parts, respectively. The factors computed here are a function of the fraction of clock time that the equipment is in operation.

94. QUE=QUE*SAVE
QUO=QUO*SAVE
QUI=QUI*SAVE
QUD=QUD*SAVE
These statements take into account the "real time" operating factor based on inherent availability to re-compute scrap LRU quantities tied up in the pipes; Equipment/Direct Support, Direct Support/General Support, General

Support/Depot and Depot/Factory, respectively.

95. QU=HE*QUE+HO*QUO+HI*QUI+HD*QUD
This statement computes the total quantity of LRU scrap tied up in the pipes, dependent upon the presence of supply at any of the support locations. If either HE, HO, HI or HD is zero (NAMELIST/L/ input as H) then there is no authorized stock at that location.

96. QFE=QFE*SAVE
QFO=QFO*SAVE
QFI=QFI*SAVE
These statements take into account the "real time" operating factor based on inherent availability to redefine the quantity of LRUs tied up in float repairs for the pipes; Equipment/Direct Support, Direct Support/General Support and General Support/Depot, respectively.

97. QME=QME*SAVP
QMO=QMO*SAVP
QMI=QMI*SAVP
QMD=QMD*SAVP
These statements take into account the "real time" operating factor based on inherent availability to redefine quantities of modules tied up in scrap for the pipes; Equipment/Direct Support, Direct Support/General Support, General Support/Depot and Depot/Factory, respectively.

98. QM=QMO+QMI+QMD+QME
This statement totals the quantity of module scrap tied up in all pipes.

99. QFME=QFME*SAVP
QFMO=QFMO*SAVP
QFMI=QFMI*SAVP
QFMD=QFMD*SAVP
These statements take into account the "real time" operating factor based on inherent availability to redefine the quantity of modules in float repairs for the pipes; Equipment/Direct Support, Direct Support/General Support, General Support/Depot and Depot/Factory, respectively.

100. QPO=QPO*SAVPP
QPI=QPI*SAVPP
QPD=QPD*SAVPP
These statements take into account the "real time" operating factor based on inherent availability to redefine the quantity of parts tied up in scrap for the pipes; Direct Support/General Support General Support/Depot, and Depot/Factory, respectively.

101. QP=QPO+QPI+QPD

This statement totals the quantity of parts tied up in scrap for all pipes.

102. IF(IBG.EQ.1) WRITE (6,BUG)
This statement prints the list of variables in NAMELIST/BUG/ when debugging of a data case is required.

J. The following section of this program determines stock supplies for LRUs, modules and parts at all supply locations using LOGAM supply rules. To enter this code the user must input AYZP=0 in the NAMELIST/L/ data deck.

The input quantity for LRUs, modules and parts are checked. If the quantity is >0.5, it is assumed that the input value is valid and the value is the number of items to be stocked. However, if the value is <0.5, it is assumed that this may not be the actual value wanted. In this case subroutine IOL is called to compute new stock values.

For LRUs a check is made to determine if stock is intended at a particular supply location. If H(i)=1, (where, i is 1 for equipment; 2 for direct; 3 for general; and 4 for depot) then stock was intended and IOL will be called; otherwise the stock value computation is ignored.

1. If (AYZP) 580, 712, 715
This statement transfers logic to either the predetermined value logic (statement 580), to the LOGAM Supply Rule logic (statement 712) or to the LOGAM Maintenance Rule logic (statement 715). The logic used predetermined depends on the value input for AYZP. AYZP<0 selects the value logic, AYZP=0 selects the LOGAM Supply Rule and AYZP>0 selects the LOGAM Maintenance Rule.

2. 715 IF (AYZP-1.) 712,713,713
This statement is used to check for the presence of force on inherent availability when using the LOGAM Maintenance Rule. Presently there is only one set of logic for Maintenance Rules, therefore transfer is made to statement 713 for both options.

3. 712 CONTINUE
This is the statement to transfer to when LOGAM Supply Rules are requested (AYZP=0).

4.
```
IF(QTE.LE.,.5.AND.H(1).EQ.1) CALL IOL
          (EDS,CKUE,QUE, QFE,QTE,ZU(1))
IF(QTO.LE..5.AND.,H(2).EQ.1) CALL IOL
          (ODS,CKUO,QUO,QFO,QTO,ZU(2))
IF(QTI.LE..5.AND.H(3).EQ.1) CALL IOL
```

```
        (DIS,CKUI,QUI,QFI,QTI,ZU(3))
IF(QTD.LE..5.AND.H(4).EQ.1) CALL IOL
        (DDS,CKUD,QUD,0.,QTD,ZU(4))
```

These statements will compute the initial LRU provisions at the Equipment, Direct Support, General Support and Depot levels, respectively; if quantities are not input (q.L.5) and stock is intended (H(i)=1) at a location.

5.
```
IF (QTME.LE..5) CALL IOL (EDS,CKME,QME,QFME,QTME,ZM(1))
IF (QTMO.LE..5) CALL IOL (ODS,CKMO.QMO,QFMO,QTMO,ZM(2))
IF (QTMI.LE..5) CALL IOL (DIS,CKMI,QMI,QFMI,QTMI,ZM(3))
IF (QTMD.LE..5) CALL IOL (DDS,CKMD,QMD,QFMD,QTMD,ZM(4))
```
   These statements will compute initial module provisions, if values are not input at the Equipment, Direct Support, General Support, and Depot levels, respectively.

6.
```
IF (QTPO.LE..5) CALL IOL (DDS,CKPO,QPO,0.,QTPO,ZP(1))
IF (QTPI.LE..5) CALL IOL (DIS,CKPI,QPI,0.,QTPI,AP(2))
IF (QTPD.LE..5) CALL IOL (DDS,CKPD,QPD,0.,QTPD,ZP(3))
```
   These statements will compute initial parts provisions, if values are not input, at Direct Support, General Support and Depot, respectively. There is no part stock at the Equipment level.

7. IF (IBG.EQ.1) WRITE (6,BUG2)
   This statement will print the variables in NAMELIST/BUG2/ when debugging of a data case is requested.

8. GO TO 580
   This statement ends the SUPPLY RULE logic and transfers around the LOGAM Maintenance Rules.

K. This section of the program computes initial provisions using LOGAM Maintenance Rules. This method of computing spares is used only when AYZP is input as a number greater than or equal to one.

1. 713 CONTINUE
   This statement is the transfer point when maintenance provisioning rules are selected (AYZP$\geq$1).

2. OFACT = ED*OTF*AYZ*8766./365
   This statement computes an operating factor that will convert the provisioning days from clock time at one Equipment installation to "real time" operation days for all ED locations. The operating time is a function of the fraction of the time the system is operational (OTF) and inherent availability (AYZ).

3. PDRE = TU*(TRC/(24.*WER/168.)+TAT(1)+TATE)*GTOT

This statement computes the provisioning days required to account for removal of LRUs at the equipment level. GTOT is the sum of all maintenance policies and TU is the total removals (A+F+FNG).

4. PDEUE1=(DSUE1+GSUE1+DUE1)*DTE
This statement computes the provisioning days for those LRUs that are evacuated from the Equipment level because there is not a maintenance policy to repair at ED.

5. PDEUE2= (DSUE2+GSUE2+DUE2)*(DTE+(TE+TC)/(24.*WER/168.))
This statement computes the provisioning days for those LRUs that are evacuated from the Equipment level to higher levels of maintenance because of their "in limbo" status. These LRUs have a maintenance policy to repair at ED, but are not repaired because of the FUE fraction.

6. PDEUE=PDEUE1+PDEUE2
This statement computes the total number of provisioning days at Equipment level due to evacuated LRUs.

7. PDSUE=(SUE1+SUE2+SUE3)*(OL(1)+SL(1))+(SUE2+SUEC)*
(TC+TE)/(24.*WER/168.)
This statement computes the provisioning days at Equipment level for scrapped LRUs. The LRUs scrapped here are those that have a maintenance policy to discard (SUE1), those "in limbo" and do not have a higher maintenance level (SUE2), and those declared not repairable after testing (SUE3).

8. PDSKUE=FGE*TC/(24.*WER/168.)
This statment computes the provisioning days at the Equipment level to test and identify false "no go" LRUs.

9. PDAU=A*(OL(1)+SL(1)+OST(1))*GTOT
This statement computes the provisioning days at the Equipment level to account for the LRU stock that is lost due to attrition. GTOT is the maintenance policy totals and is used here to prevent a value being computed for PDAU when the user fails to input a maintenance policy(s).

10. PDRUE=TUFTE*(TC+TE+TER)/(24.*WER/168.)
This statement computes the provisioning days to repair LRUs at the Equipment level. TUFTE is the flow of LRUs through the checkout and test facilities that were declared repairable.

11. PDUE=(PDRE+PDEUE+PDSUE+PDAU+PDSKUE+PDRUE)*OFACT
This statement sums the provisioning days at the Equipment level for LRUs removed, evacuated, scrapped, and repaired. OFACT is then used to compute the quantity of initial LRU provisions as a function of "real time" operation at all ED locations.

12. PDEUO1=(GSUO1+DUO1)*(DTO+TC/(24.*WOR/168.)+TAT(2))
This statement computes the provisioning days for those LRUs that are evacuated from the Direct Support level to higher maintenance levels because there is not a maintenance policy to repair at DS.

13. PDEUO2=(GSUO2+DUO2)*(DTO+(TC+TF)/(24.*WOR/168.)+TAT(2))
This statement computes the provisioning days for those LRUs that are evacuated from Direct Support to higher levels of maintenance because of their "in limbo" status. These LRUs have a maintenance policy to repair at DS, but are not repaired because of the FUE fraction.

14. PDEUO=PDEUO1+PDEUO2
This statement computes the total number of provisioning days at Direct Support due to evacation of LRUs to a higher maintenance level.

15. PDSUO=(SUO1+SUO2+SUO3)*(TC/(24.*WOR/168.)+TAT(2)+OL(2)+
SL(2)+OST(2))+(SUO4+SUO5)*((TC+TF)/(24.*WOR/168.)
+TAT(2)+OL(2)+SL(2)+OST(2))
This statement computes the provisioning days for LRUs scrapped at Direct Support.

16. PDSKUO=FGO*(TC/(24.*WOR/168.)+TAT(2))
This statement computes the provisioning days at Direct Support to test and identify false "no go" LRUs.

17. PDRUO=TUFTO*((TC+TF+TFR)/(24.*WOR/168.)+TAT(2))
This statement computes the provisioning days at Direct Support to repair LRUs. TUFTO is the flow of LRUs through the DS test facility that were declared repairable.

18. PDUO=(PDEUO+PDSUO+PDSKUO+PDRUO)*OFACT
This statement sums the provisioning days at Direct Support for LRUs that are evacuated, scrapped, tested as false "no go's", and repaired. OFACT is then used to compute the quantity of initial LRU provisions as a function of "real time" operations of all ED locations.

19. PDEUI=DUI*((TC+TI)/(24.*WIR/168.)L+TAT(3)+DTI)
This statement computes the provisioning days at General Support for LRUs evacuated to a higher maintenance level (Depot) because there is not a maintenance policy to repair at GS.

20. PDSUI=(SUI1+SUI2)*(TC*TI)/(24.*WIR/168.)+
TAT(3)+OL(3)+SL(3)+OST(3))
This statement computes the provisioning days for LRUs scrapped at General Support.

21. PDSKUI=FGI*(TC/(24.*WIR/168.)+TAT(3))
This statement computes the provisioning days at General Support required to test and identify false "no go" LRUs.

22. PDRUI=TUFTI*((TC+TI+TIR)/(24.*WIR/168.)*OFACT
This statement computes the provisioning days at General Support to repair LRUs. TUFTI is the flow of LRUs through the test facility that were declared repairable.

23. PDUI=(PDEUI+PDSUI+PDSKUI+PDRUI)*OFACT
This statement sums the provisioning days at General Support for LRUs that are evacuated, scrapped, tested as false "no go's", and repaired. OFACT is then used to compute the quantity of initial LRU provisions as a function of "real time" operations at all ED locations.

24. PDSUD=SUD*((TC+TD)/(24.*WDR/168.)+
TAT(4)+OL(4)+SL(4)+OST(4))
This statement computes the provisioning days for LRUs scrapped at the Depot.

25. PDSKUD=FGD+(TC/(24.*WDR/168.)+TAT(4)+STAT)
This statement computes the provisioning days at the Depot for checkout, turn around, and shipping of false "no go" LRUs.

26. PDRUD=TUFTD*((TC+TD+TDR)/(24.*WDR/168.)+
TAT(4)+STAT)
This statement computes the provisioning days at the Depot required for checkout, test, and repair of LRUs. TUFTD is the flow of LRUs through the test facility that were declared repairable.

27. PDUD=(PDSUD+PDSKUD+PDRUD)*OFACT
This statement sums the provisioning days for LRUs at the Depot that are scrapped, tested for false "no go's", and repaired. OFACT is then used to compute the quantity of intial provisions as a function of "real time" operations at all ED locations.

28. TOTPDU=PDUE+PDUO+PDUI+PDUD
This statement sums the initial LRU provisions from all stock locations.

29. PDEME=TMFEO*(TAT(1)+DTE)
This statement computes provisioning days at the Equipment level for modules evacuated to higher echelons of maintenance.

30. PDSME=TSME*(TAT(1)+OL(1)+SL(1)+OST(1))
This statement computes the provisioning days at the Equipment level for scrapped modules.

31. PDME=(PDEME+PDSME)*OFACT
This statement sums the provisioning days for evacuations and scrap modules at the Equipment level. OFACT is used to compute the initial module provisions as a function of "real time" operations at all ED locations.

32. PDEMO=(GSMO+DMO)*(TAT(2)+DTO)
This statement computes the provisioning days for modules evacuated from Direct Support to higher echelons of maintenance.

33. PDSMO1=SMO1*(TMO/(24.*WOR/168.))
This statement computes the provisioning days for testing modules that are scheduled for repair at Direct Support but were declared not repairable (scrap).

34. PDSMO2=TSMO*(TAT(2)+OL(2)+SL(2)+OST(2))
This statement computes the provisioning days for replacing all scrapped modules at Direct Support.

35. PDSMO=PDSMO1+PDSMO2
This statement computes the provisioning days at Direct Support for the testing and replacement of scrapped modules.

36. PDRMO=TMFO*((TMO+TMOR)/(24.*WOR/168.)+TAT(2))
This statement computes the provisioning days at Direct Support for testing and repairing modules.

37. PDMO=(PDEMO+PDSMO+PDRMO)*OFACT
This statement sums the provisioning days for modules that are evacuated, scrapped, and repaired at Direct Support. OFACT is used to compute the quantity of initial module provisions at DS as a function of "real time" operations at all ED locations.

38. PDEMI=DMI*(TAT(3)+DTI)
This statement computes the provisioning days for modules that are evacuated from General Support to a higher maintenance echelon (Depot).

39. PDSMI1=SMI1*(TMI/(24.*WIR/168.)+
TAT(3)+OL(3)+SL(3)+OST(3))
This statement computes the provisioning days at General Support for modules that are scheduled for repair at GS but after testing are declared not repairable (scrap).

40. PDSMI2=(SMI2+SMI3)*(TAT(3)+SL(3)+OST(3))
This statement computes the provisioning days at General Support for modules that are scrapped because the maintenance policy is to discard or evacuate to a higher maintenance level. These modules are not charged test time

(TMI).

41. PDSMI=PDSMI1+PDSIM2
This statement computes the provisioning days for all modules scrapped at General Support.

42. PDRMI=TMFI*((TMI+TMIR)/(24.*WIR/168.)+TAT(3))
This statement computes the provisioning days for modules tested and repaired at General Support.

43. PDMI=(PDEMI+PDSMI+PDRMI)*OFACT
This statement sums the provisioning days at General Support for modules evacuated, scrapped, and repaired. OFACT is used to compute the initial module provisions at GS as a function of "real time" operations at all ED locations.

44. PDSM1=SMD1*(TMD/(24.*WDR/168.)+TAT(4)+OL(4)+SL(4)+OST(4))
This statement computes the provisioning days at the Depot for modules that are tested and declared not repairable (scrap).

45. PDSM2=SMD2*(TAT(4)+OL(4)+SL(4)+OST(4))
This statement computes the provisioning days at the Depot for modules that are scrapped due to a maintenance "discard" policy and/or after screening for false "no go's".

46. PDSMD=PDSMD1+PDSMD2
This statement computes the provisioning days for all module scrap at the Depot.

47. PDRMD=TMFD*((TMD+TMDR)/(24.*WDR/168.)+TAT(4)+STAT)
This statement computes the provisioning days at the Depot for modules tested and repaired.

48. PDMD=(PDSMD+PDRMD)*OFACT
This statement sums the provisioning days at the Depot for modules that are scrapped and repaired. OFACT is used to compute the quantity of initial module provisions at the Depot as a function of "real time" operations at all ED locations.

49. TOTPDM=PDME+PDMO+PDMI+PDMD
This statement sums the initial module provisions of all stock locations into a system total.

50. PDPO=PDRMO*OFACT
PDPI=PDRMI*OFACT
PDPD=PDRMD*OFACT
These statements compute the initial provisions for parts at Direct Support, General Support, and Depot, respectively. The provisioning days for module repair is used to provision for parts since each time a module is repaired, a part is

required.

51. TOTPDP=PDPO+PDPI+PDPD
This statement sums the initial provisions for parts at Direct Support, General Support, and Depot, into a total system provisioning.

The following statements distribute the LRU, module, and part stock according to the authorized stock points. The H matrix, which is input, determines the presence or absence of stock at a stock level. An H value of one (1) authorizes stock at a location. Computed provisioning quantities are distributed equally over all stock points at a particular stock level and rounded to whole units as a function of the "Z" parameter. Any provisions "borrowed" or "left over" from the rounding function will be accounted for in the distribution at the next higher authorized stock level. When stock is not authorized at a particular level the stock is distributed at the next lowest authorized level. The equipment level stock is an exception since there is not a lower stock level. In this case the stock is "left over" and will be distributed at the next higher authorized level.

52.
```
QTE=EDS*AINT((PDUE+(1.-H(2))*PDUO+(1.-H(2)*
     (1.-H(3))*PDUI+(1.-H(2))*(1.-H(3))*
     (1.-H(4))*PDUD/D(EDS)+ZU(1))*H(1)
```
This statement, if H(1)=1, will distribute the LRU stock computed at the Equipment level (PDUE) equally over all ED stock points. The stock computed at the next higher echelon(s) for which stock is not authorized will also be distributed here. The distributed stock is rounded (ZU) to whole units.

53. DQ=PDUE-QTE
This statement computes the quantity of LRU stock "borrowed" or "left over" after distribution at the ED level. When the ZU fraction causes a round up of the stock quantity, stock is borrowed from the next higher echelon. Also the stock computed for a higher echelon that does not have stock authorization will be distributed at ED and this stock is also concidered borrowed. The left over stock occurs when the computed quantity is rounded off because of the ZU fraction.

54.
```
QTO=ODS*AINT((PDUO+DQ+(1.-H(3))*PDUI+(1.-H(3)*
     (1.-H(4)*PDUD/D(ODS)+ZU(2))*H(2)
```
This statement, if H(2)=1, will distribute the LRU stock computed at the Direct Support level (PDUO) equally over all OD stock points. The stock computed at OD will be adjusted by the "borrowed" or "left over" stock quantity from the ED level before the distribution is made. The stock computed at the next higher echelon(s) for which stock is not

authorized will also be distributed here.

55. IF(QTO.LT.0.)QTO=0.
This statement will set the LRU stock quantity at Direct Support to zero when the quantity of stock to be distributed at DS is less than the quantity "borrowed" from the ED level.

56. DQ=PDUO-QTO+PDUE-QTE
This statement computes the quantity of LRU stock either "borrowed" or "left over" when stock was distributed at the ED and OD levels.

57.
```
QTI=DIS*AINT(PDUI+DQ+(1.-H(4))*PDUD/D(DIS)+
     ZU(3)))*H(3)
```
This statement, if H(3)=1, will distribute the LRU stock computed at the General Support level (PDUI) equally over all DI stock locations. The stock borrowed or left over from distributions at ED and OD is used to adjust the computed quantity before a distribution is made. The stock computed at Depot will also be distributed here if stock is not authorized at the Depot.

58. IF(QTI.LE.0.)QTI=0.
This statement will set the LRU stock quantity at General Support to zero when the quantity of stock to be distributed at GS is less than the quantity "borrowed" at the lower stock levels.

59. DQ=PDUI-QTI+PDUE-QTE+PDUO-QTO
This statement computes the quantity of LRU stock "borrowed" or "left over" when stock was distributed at the GS, DS, and E levels.

60. QTD=DDS*AINT((PDUD+DQ)/D(DDS)+ZU(4))*H(4)
This statement, if H(4)=1, will distribute LRU stock computed at the Depot level (PDUD) equally over all Depot stock locations. The stock "borrowed" or "left over" from distributions at E, DS, and GS is used to adjust the computed quantity before a distribution is made.

61. IF(QTD.LE.0.)QTD=0.
This statement will set the LRU stock distributed at the Depot to zero if the quantity "borrowed" at the lower echelon distributions is greater than the computed quantity.

62.
```
QTME=EDS*AINT((PDME+(1.-H(2)*PDMO+(1.-H(2)*(1.-H(3))+
     PDMI+(1.-H(2))*(1.-H(3))*(1.-H(4))*PDMD/D(EDS)+
     ZM(1))*H(1)
```
This statement distributes the computed module stock (PDME) equally over all ED stock locations. The logic for distributing modules is the same as the distribution for

LRUs.

63. DQ=PDME-QTME
This statement computes the quantity of module stock either "borrowed" on "left over" in the Equipment level distributions.

64. QTMO=ODS*AINT((PDMO+DQ+(1.-H(3))*PDMI+(1.-H(3))
*(1.-(4))*PDMD/D(ODS)+ZM(2))*H(2)
This statement distributes the computed module stock (PDMO) at Direct Support equally over all OD stock locations.

65. IF(QTMO.LE.0.)QTMO=0.
This statement sets the module stock distributions at Direct Support to zero when the quantity "borrowed" from the Equipment distribution is greater than the quantity to be distributed at OD.

66. DQ=PDMO-QTMO+PDME-QTME
This statement computes the quantity of module stock either "borrowed" or "left over" when distributions where made at the DS and E levels.

67. QTMI=DIS*AINT((PDMI+DQ+(1.-H(4))*PDMD)/D(DIS)+ZM(3))*H(3)
This statement distributes the computed module stock (PDMI) at General Support equally over all DI stock locations.

68. IF(QTMI.LE.0)QTMI=0.
This statement sets the module stock distribution at General Support to zero when the quantity "borrowed" for the Equipment and Direct Support locations is greater than the quantity to be distributed at GS.

69. DQ=PDMI-QTMI+PDME-QTMEPDMO-QTMO
This statement computes the quantity of module stock either "borrowed" or "left over" after distributions are made at the Equipment, Direct Support, and General Support stock locations.

70. QTMD=DDS*AINT((PDMD+DQ)/D(DDS)+ZM(4))*H(4)
This statement distributes the computed module stock (PDMD) at the Depot equally over all DD stock locations.

71. IF(QTMD.LE.0.)QTMD=0.
This statement sets the module stock distribution at the Depot to zero when the stock "borrowed" for distributions at E, DS, and GS is greater than the quantity computed (PDMD) for the Depot.

72. QTPO=ODS*AINT((PDPO+(1.-H(3))*PDPI+(1.-H(3))*
(1.-H(4))*PDPD)/D(ODS)+ZP(1))*H(2)
This statement distributes the computed part stock (PDPO) at

Direct Support equally over all DS stock locations. The logic for distributing part stock is the same as the distribution for LRU stock.

73. DQ=PDPO-QTPO
This statement computes the quantity of part stock either "borrowed" or "left over" when distributions were made at Direct Support.

74. QTPI=DIS*AINT((PDPI+DQ+(1.-H(4))*PDPD)/
        D(DIS)+ZP(2))*H(3)
This statement distributes the computed part stock (PDPI) at General Support equally over all GS stock locations.

75. IF(QTPI.LE.0.)QTPI=0.
This statement sets the part stock distribution at General Support to zero when the stock "borrowed" for distribution at DS is greater then the quantity to be distributed at GS.

76. DQ=PDPI-QTPI+PDPO-QTPO
This statement computes the quantity of part stock either "borrowed" or "left over" when distributions were made at DS and GS.

77. QTPD=DDS*AINT((PDPD+DQ)/D(DDS)+ZP(3))*H(4)
This statement distributes the computed part stock (PDPD) at Depot equally over all DD stock locatons.

78. IF(QTPD.LE.0)QTPD=0.
This statement sets the part stock distribution at Depot to zero when the stock "borrowed" for distribution at DS and GS is greater than the quantity to be distributed at the Depot level.

79. 580 IF(IBG.EQ.1) WRITE (6,BUG6)
This statement prints the initial provision values when this debugging a data case. Statement 580 is entered after provisions are either predetermined, computed by the LOGAM Supply Rule, or computed by the LOGAM Maintenance Rule.

L. This section of the program computes the expected value demands for stock based on inherent availability, computes operational availability, and redefines the supply quantities based on operational availability. This code is entered after supply quantities have either been predetermined, computed by the LOGAM Supply Rules, or computed by the LOGAM Maintenance Rules.

1. RQU=(QUE+QFE)*H(1)+(QUO+QFO)

```
        *AB(H(1)+H(2)+H(3))+QUD*
        AB(H(1)+H(2)+H(3)+H(4))
RQM=QMO+QFMO+QMI+QFMI+QMD+QFMD+QME+QFME
RQP=QPO+QPI+QPD
```

These statements compute the expected value demand by summing scrap and float quantities for LRUs, modules and parts, respectively.

2.

```
QT=QTE+QTO+QTI+QTD
QTM=QTMO+QTMI+QTMD+QTME
QTP=QTPO+QTPI+QTPD
```

These statements compute the total requirement for initial spares for LRUs, modules and parts, respectively by summing quantities of stock at each echelon.

3.

```
OR=AMOD(ABS(AYZP),1.)
IF(OR.LT..5)OR=0.
```

These statements set the override stock control factor. The fractional part of AYZP is the override factor. The override factor forces the operational availability to be at a minimum a set percent of the inherent availability.

4.

```
DO 443 JOR=1,2500
```

This statement loops through the operational availability code until the availability goal is met. This logic assumes that the availability goal can be met within 2500 passes.

5.

```
QYU=DEF(RQU+P*DEF(RQM+PP*DEF(RQP,QTP,1.)/D(P),
        QTM,1.),QT,1.)/SAVE
```

This statement computes the Back Order Quantity (BOQ) for unfilled items from the warehouse. The BOQs are computed by function DEF which is based on a poisson distribution of demand per resupply interval. The statement is a nest of three calls to DEF. The first call calculates the shortage of parts (DEF(RQP,QTP,1.)), multiplies this by the number of part types (PP), and divides by the number of module types (D(P)) to determine how the shortage of part types will affect the demand on each module type. The shortage of modules increases the demand for modules on the next call (DEF(RQM+PP*___,QTM,1.)). The shortage of modules in turn increases the demand for LRUs in the third and final call (DEF(RQU+P*___,QT,1). Dividing by SAVE converts the results to the shortage of LRUs at a single location.

6.

```
DEN1=1.-E*EE*OTF*TRC-SMF-FNG*OTF*TRC
```

This statement computes the fraction of time that the system is up and operational after adjustments are made for down-time due to service demands (TRC) and scheduled maintenance (SMF). Since all the variables in this statement are input, the statement should be evaluated before starting the DO 443 Loop.

7. IF(DEN1)7161,7162,7161
This statement will allow the operational availability to remain at zero (AYZ=0) if the fraction of up-time is a zero by transferring to statement 7162. When up-time is non zero AYZ will be computed. Warning, a negative value for DEN1 will result in an operational availability (AYZ) that is greater than 100 percent.

8. 7162 AYZ=0.
This statement initializes operational availability.

9. 7161 CONTINUE
OTFN=OTF/DEN1
Transfer is made to these statements when there is a non zero value for DEN1. The fraction of real time for equipment in operation (OTF) is modified by the up-time fraction (DEN1).

10. AYZ=EE/(EE+OTFN*(QYZ+QYU))
This statement computes operational availability from the modified operational equipment fraction (OTFN) and the sum of the down quantity (QYZ) and the back order quantity (QYU).

11. AYZOS=SPOL(AYZ,FN,EE)
This statement modifies the operational availability in subroutine SPOL to account for redundant equipment or built in spares.

12. IF(AYZOS.GE.(AYZIS*OR).OR.OR.LT..5)GO TO 993
This statement checks to determine if a force on inherent availability was requested (OR>.5). If a force was requested a check is made to determine if the force has been achieved. When the force is achieved, a transfer is made outside of the DO 443 Loop to statement 993. Otherwise logic stays in the DO 443 loop where additional quantities are added and another pass through the operational availability equation is made.

13.
```
    IU=4
    IF(H(3).EQ.1)IU=3
    IF(H(2).EQ.1)IU=2
    IF(H(1).EQ.1)IU=1
    GO TO (953,963,973,983),IU
953 QTE=QTE+EDS
    QT=QT+EDS
    GO TO 443
963 QTO=QTO+ODS
    QT=QT+ODS
    GO TO 443
973 QTI=QTI+DIS
    QT=QT+DIS
```

```
    GO TO 443
983 QTD=QTD+DDS
    QT=QT+DDS
```

These statements are evaluated only when a force on inherent availability is to be achieved. Stock quantities are added to previously evaluated LRU provisions by adding one item for each supply point. The level of the supply point for which quantities are added is determined by the lowest echelon for which stock is permitted. For instance if stock is allowed at all echelons except at the Equipment level (H(1)=0), then ODS items of stock will be added to the quantity of stock (QTO) at Direct Support.

14. 443 CONTINUE

This statement is the end of the operational availability DO-LOOP. If the force on inherent availability has not been achieved within 2500 attempts then program logic will automatically drop to the next statement.

15. 993 CONTINUE

Transfer is made to this statement once the operational availability level has been reached or 2500 attempts have been made.

16. DO 943 I=1,NA

```
943 IF(TAYZ(I).EQ.1.) CAYZ(I)=CAYZ(I)*(AYZOS**REPEAT)
```

These statements tally operational availabilities for subsystems of LRUs. The system availability is stored in the first location of CAYZ. LRU subsystem availabilities are determined by the location of the TAYZ inputs within the data deck. CAYZI is used to print subsystem availabilities on the case total output page.

17. SAVE=ED*OTF*AYZ

```
SAVP=SAVE/D(P)
SAVPP=SAVE/D(PP)
```

These statements compute conversion factors for LRUs, modules, and parts, respectively, as a function of operational availability (AYZ). The factors are used to convert quantities and scrap rates from "clock time" values per installation to "real time" (OTF) values for total installations (ED).

18. SAVEEB=SAVE/D(ED)

```
SAVEOB=SAVE/D(OD)
SAVEIB=SAVE/D(DI)
SAVEDB=SAVE/D(DD)
```

These statements compute factors based on operational availability for the effective number of installations supported by the Equipment, Direct, General and Depot facilities, respectively.

19. SUT=SAVE*TSU
SUTF=SAVE*TSLF
SUTD=SAVE*TSLD
These statements use the conversion factor SAVE to compute "real time" scrap rates of LRUs at all support levels, Field support, and Depot support, respectively.

20. TSM=TSMO+TSMI+TSMD+TSME
U6=TSM-TSMD
These statements compute "clock time" scrap rates of modules from one equipment installation at all support levels and Field support, respectively.

21. SMTF=SAVP*U6
SMTD=SAVP*TSMD
SMT=SAVP*TSM
These statements use the conversion factor SAVP to compute "real time" scrap rates for Field support, Depot support, and total support respectively.

22. TSP=TSPO+TSPI+TSPD
U7=TSP-TSPD
These statements compute "clock time" scrap rates of parts from one equipment installation at all support levels, and Field support, respectively.

23. SPTF=SAVPP*U7
SPTD=SAVPP*TSPD
SPT=SAVPP*TSP
These statements use the conversion factor SAVPP to compute 'real time" scrap rate of parts at Field support, Depot support, and all support levels, respectively.

24. QUDH=QUDH*SAVE
QMDH=QMDH*SAVP
QPDH=QPDH*SAVPP
These statements compute quantities of stock tied up in discretionary procurement holding time as a function of total deployment and "real time" operation for LRUs, Modules and Parts, respectively.

25. YR8=8766.*YR
ONTIME=YR8*SAVE
These statements convert the years of operation and maintenance into clock hours (YR8) and then to "real time" operational hours.

26. IF(IBG.EQ.1)WRITE(6,BUG7)
This statement prints the list of variables for NAMELIST/BUG7/ when debugging of a data case is required.

27. QTD=QTD+AINT(QUDH)

This statement adds the quantity of LRU stock at the Depot with the quantity of LRU stock needed because of procurement holding time (QUDH) to compute initial LRU provisions at the Depot.

28. QT=QTE+QTO+QTI+QTD
This statement totals the initial LRU provisions from all stock levels.

29. QUA=EDEE+QT
This statement computes the initial buy quantity for LRUs. This includes those operational at all equipment installations (EDEE) and the initial provisions (QT).

30. U4=QTE+QTO+QTI+EDEE
This statement computes the initial buy quantity for LRUs located at the field facilities.

31.
```
QQC=YR8*SUT
QQCF=YR8*SUTF
QQCD=YR8*SUTD
```
These statements compute the quantities of consumed LRUs over the O&M lifecycle for total maintenance, maintenance at all Field levels, and Depot maintanance, respectively.

32. QU=QU+QUDH
This statement computes the quantity of LRUs needed to cover the reprocurment cycle by adding the quantities tied up in the reprocurement holding time (QUDH) to the quantity of LRUs tied up in the scrap pipelines. The QU of (QU+QUDH) was computed based on inherent availability where QUDH is based on operational availability.

33. PURX=QUA/YPF
This statement computes the LRU hourly production rate.

34.
```
QC=AMAX1(0.,QQC-QT)
U19=QT-QTD
QCF=AMAX1(0.,QQCF-U19)
QCD=AMAX1(0.,QQCD-QTD)
```
These statements compute the quantity of LRUs consumed over and above the intital provisions for total system, Field echelons, and Depot, respectively. When initial provisions are greater than scrap quantities the consumed quantities are set to zero.

35.
```
     IF(QC.LE.O.)GO TO 5084
     GO TO 5085
5084 QCD=0.
     QCF=0.
     GO TO 5086
5085 CONTINUE
```

```
    IF (QCF.LE.O.)QCD=QC
    IF (QCD.LE.O.)QCF=QC
5086 CONTINUE
```

These statements set the quantity of LRU stock over and above the scrap quantities that are required as safety stock. If safety stock is not required (QC>0) the safety deltas at Depot (QCD) and field echelons (QCF) are set to zero. Otherwise the deltas are set to QC unless quantities were previously evaluated at these echelons.

36.
```
PURY=PUR
IF(PURY.LE..1E-19)PURY=PURX
PURZ=2.*SUT
IF(PURY.LT.PURZ)PURY=PURZ
```

These statements set the LRU hourly production rate (PURY). If the input rate (PUR) or the computed rate (PURX) is less than twice the scrap rate (SUT), hourly production rate is set to twice the LRU scrap rate.

37.
```
QB=QU/(1.-(SUT/PURY))
IF(QB.LT.QMU)QB=QMU
IF(PURZ.GT.PURX)QB=QC
```

These statements compute the lot size for LRU reorder buys.

38.
```
PMRY=PMR
```

This statement sets the hourly production rate for modules to the rate input.

39.
```
QM=QM+QMDH
```

This statement computes the quantity of modules needed to cover the reprocurement cycle by adding the quantities tied up in the procurement holding time (QMDH) to the quantity of modules tied up in the scrap pipelines.

40.
```
QTMD=QTMD+AINT(QMDH)
QTM=QTMO+QTMI+QTMD+QTME
```

These statements recompute the initial provisions for module stock at the Depot and at all support points, respectively, by adding the quantities tied up in discretionary procurement holding time.

41.
```
QQCM=YR8*SMT
QCM=AMAX1(0.,QQCM-QTM)
QQCMF=YR8*SMTF
QQCMD=YR8*SMTD
```

These statements compute the consumed Module quantities over the O&M lifecycle for total maintenance (QQCM), maintenance at Field echelons (QQCMF), and maintenance at Depot (QQCMD). QCM is the Modules consumed over and above the initial provisions.

42.
```
U8=QTM-QTMD
```

This statement computes the initial provisions for modules at the Field supply points.

43.
```
          IF(QCM.LE.0.)GO TO 5087
          GO TO 5088
     5087 QCMD=0.
          QCMF-0.
          GO TO 5089
     5088 CONTINUE
          QCMF=AMAXI(0.,QQCMF-U8)
          QCMD=AMAX1(0.,QQCMD-QTMD)
     5089 CONTINUE
          IF(QCMF.LE.0.)QCMD=QCM
          IF(QCMD.LE.0.)QCMF=QCM
```
These statements set the quantity of Modules that are required as safety stock at the Depot (QCMD) and the Field echelons (QQMF).

44.
```
QMA=QUA+QTM
```
This statement computes the initial buy quantity for modules. This includes a Module for each initial LRU buy (QUA) plus the initial stock provisions.

45.
```
PMRX=QMA/YPF
IF(PMRY.LE..1E-19)PMRY=PMRX
PMRZ=.2.*SMT
IF(PMRY.LT.PMRZ)PMRY=PMRZ
```
These statements set the hourly production rates for modules. If the input rate (PMR) or the computed rate (PMRX) is less than twice the Module scrap rate (SMT), hourly production rate is twice the scrap rate.

46.
```
QBM=QM/(1.-(SMT/PMRY)
IF(QBM.LT.QMM)QBM=QMM
IF(PMRZ.GT.PMRX)QBM=QCM
```
These statements compute the reorder buy lot for modules.

47.
```
PPRY=PPR
```
This statement sets the production rate for Parts to the input value.

48.
```
QP=QP+QPDH
```
This statement computes the quantity of parts needed to cover the reprocurement cycle by adding the quantities tied up in the procurement holding time (QPDH) to the quantity of parts tied up in the scrap pipelines.

49.
```
QTPD=QTPD+AINT(QPDH)
QTP=QTPO+QTPI+QTPD
```
These statements recompute initial provisions parts by adding in the stock quantity tied up in procurement holding time to the provisions at Depot and provisions at all

echelons, respectively.

50. QPA=QMA+QTP
This statement computes the initial buy quantity for parts. This includes one part for each module buy plus the initial provisions.

51. QQCP=YR8*SPT
This statement computes the parts stock required for maintenance (consumed parts) during the O&M lifecycle.

This statement computes the quantity of parts needed to cover the reprocurement cycle by adding the quantities tied up in the procurement holding time (QPDH) to the quantity of parts tied up in the scrap pipelines.

52.
```
PPRX=QPA/YPF
IF(PPRY.LE..1E-19)PPRY=PPRX
PPRA=2.*SPT
IF(PPRY.LT.PPRA)PPRY=PPRA
```
These statements set the hourly production rate for parts. If the input production rate (OPR) or the computed rate (PPRX) is less than twice the part scrap rate (SPT); hourly parts production rate is then twice the scrap rate.

53. QCP=AMAX1(0.,QQCP-QTP)
This statement computes the quantity of consumed parts during O&M over and above the initial provisions for the total system.

54. U9=QTP-QTPD
This statement computes the initial provision for parts at the Field echelons.

55.
```
QQCPF=YR8*SPTF
QQCPD=YR8*SPTD
```
These statements compute the quantity of consumed parts during O&M at the Field and Depot facilities, respectively.

56.
```
     IF(QCP.LE.0.)GO TO 5090
     GO TO 5091
5090 QCPD=0.
     QCPF=0.
     GO TO 5092
5091 CONTINUE
     QCPF=AMAX1(0.,QQCPF-U9)
     QCPD=AMAX1(0.,QQCPD-QTPD)
5092 CONTINUE
     IF(QCPF.LE.0.)QCPD=QCP
     IF(QCPD.LE.0.)QCPF=QCP
```
These statement compute the quantity of parts over and above the initial provisions that are consumed during the O&M

phase at the Field (QCPF) and Depot (QCPD) maintenance facilities.

57.
```
   QBP=QP/(1.-(SPT/PPRY))
   IF(QBP.LT.QMP)QBP=QMP
18 IF(PPRA.GT.PPRX)QBP=QCP
```
These statements set the reorder lot size for parts. The lot size can never be less than the minimum lot size (QMP) input.

58.
```
AQP=AB(QTP)
```
This statement computes a multiplier whose value will be one (1) or zero (0) depending on presence or absence of stocked parts (QTP). This variable is used later to cost parts.

59.
```
UCUP=CUP
UCUR=CUP
UCMP=CMP
UCMR=CMP
UCPP=CPP
UCPR=CPP
```
These statements initialize cost factors from input values for LRU's, modules, and parts.

M. This section of the program computes work load factors for maintenance locations.

1.
```
AME=SAVEEB*TU*TRC
```
This statement computes the manhours to fault isolate, remove, test, and replace a LRU at one equipment location.

2.
```
SAME=SAME+AME*REPEAT
```
This statement accumulates the manhours to fault isolate, remove, replace, and test all LRUs at one Equipment location per clock hour.

3.
```
ESME=ETE*SAME*(1.+FE)
```
This statement adds the manhours for self-support of type V test equipment to SAME and controls posting out of accumulated work demands. ETE=1, posts work demands, ETE=0 removes work demands.

4.
```
DSIM=EREI*ED*168.*(ESME/D(WMR))+
        ((1.-EREI)*ED*AINT((168.*
        ESME/D(WMR))+ZFL))
```
This statement computes the number of unscheduled maintenance personnel to do LRU fault isolation, remove, replace and test per clock hour at the equipment level for

all maintenance locations.

5.
```
ESI=(ETEI*ED*168.*(1.-RF)*
        (ESME/D(WMT)))+((1.-ETEI)*ED*
        AINT((168.*(1.-RF)*ESME/D(WMT))+ZFL))
```
This statement computes the demand for Type V test equipment to fault isolate and test LRUs at the equipment level for all equipment level maintenance locations.

6.
```
AEY=SAVEEB*((TUFEC*TC)+(TUFTE*TE))
AERY=SAVEEB*((TUFTE*TER))
```
These statements compute manhours to test and check out, and manhours to repair a LRU at one equipment maintenance location, respectively.

7.
```
AOY=(SAVEOB*((TUFOC*TC)+(TUFTO*TF)+(TMFO*TMO)))
        +((QUA/D(OD))*2.*TMWO*TOMW)
AORY=(SAVEOB*((TUFTO*TFR)+(TMFO*TMOR)))+
        ((QUA/D(OD))*TMWO*TMOD)
```
These statements compute the manhours to test and repair, respectively, one LRU from each material system through one Direct Support facility.

8.
```
AIY=(SAVEIB*((TUFIC*TC)+(TUFTI*TI)+(TMFI*TMI)))+
        ((QUA/D(DI)*2.*TMWI*TIMW)
AIRY=(SAVEIB*((TUFTI*TIR)+(TMFI*TMIR)))+
        ((QUA/D(DI))*TMWI*TMID)
```
These statements compute the manhours to test and repair, respectively, one LRU from each material system through one General Support facility.

9.
```
ADY=(SAVEDB*((TUFDC*TC)
        +(TUFTD*TD)+(TMFD*TMD)))+
        ((QUA/D(DD))*2.*TMWD*TDMW)
ADRY=(SAVEDB*((TUFTD*TDR)+(TMFD*TMDR)))+
        ((QUA/D(DD))+TMWD*TMDD)
```
These statements compute the manhours to test and repair, respectively, one LRU from each material system through one Depot facility.

10.
```
SAEY=SAEY+AEY*REPEAT
```
This statement accumulates the test manhours at one equipment level for all LRU removals in a materiel system.

11.
```
BSAEY=SAEY
```
This statement stores the value of SAEY in COMMON/SUPIN/for output in subroutine SUPI.

12.
```
ESUY=ETI*SAEY*(1.+FI)
```
This statement adds the type I test equipment manhour demand for self-support to the LRU test demand (SAEY) at the equipment level.

13. ESU=(EVET*ED*168.*(ESUY/D(WE))+((1.-EVET)*ED
AINT((168.(ESUY/D(WE)))+ZFL))
This statement computes the expected value of the service demand for Type I test equipment at the Equipment level for all deployed equipments.

14. ESUM=(EVEM*ED*168.*(ESUY/D(WEM)))+((1.-EVEM)*
ED*AINT((168.*ESUY/D(WEM)))+ZFL)
This statement computes the expected value demand for test manpower at the equipment level for all deployed equipments.

15. SAERY=SAERY+(AERY*REPEAT)
This statement accumulates repair manhours at the Equipment level for LRUs from all materiel systems.

16. BSAERY=SAERY
This statement stores the value of SAERY into COMMON/SUPIN/ for output in subroutine SUPI.

17. ESURY=ETI*SAERY
This statement controls posting out of the accumulated repair work demand for Type I test equipment at Equipment Level service channels. ETI=1 is input for posting, otherwise, ETI=0 is input.

18. ESUR=(EVER*ED*168.*(ESURY/D(WER)))+((1.-EVER)*
ED*AINT((168.*(ESURY/D(WER)))+ZFL)
This statement computes the expected value demand on repair manpower at all equipment levels.

19. SAOY=SAOY+(AOY*REPEAT)
This statement accumulates test manhours for LRUs through one Direct Support facility as a function of all materiel system removals.

20. DSUY=ETI*SAOY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at the Direct Support facility. To post this work demand, ETI=1 is input, otherwise, ETI=0 is input.

21. DSU=(EVOT*OD*168.*(DSUY/D(WO)))+((1.-EVOT)*OD*
AINT((168.*(DSUY/D(WO))).+ZFL))
This statement computes the expected value of the service demand for Type I test equipment at all Direct Support maintenance locations.

22. DSUM=(EVOM*OD*168.*(DSUY/D(WOM)))+(1.-EVOM)*OD*
AINT((168.*(DSUY/D(WOM)))+ZFL))
This statement computes the expected value demand on test manpower for LRUs through all Direct Support maintenance

locations.

23. SAORY=SAORY+(AORY*REPEAT)
This statement accumulates repair manhours for all LRUs through a Direct Support facility.

24. DSURY=ETI*SAORY
This statement controls posting out of the accumulated repair work demand for Type I test equipment at the Direct Support service channel. ETI=1 is input for posting, otherwise, ETI=0 is input.

25.
```
DSUR=(EVOR*OD*168.*(DSURY/
        D(WOR)))+((1.-EVOR)*OD*
        AINT((168.*(DSURY/D(WOR)))+ZFL))
```
This statement computes the expected value demand on repair manpower at all Direct Support maintenance locations.

26.
```
SAIY=SAIY+(AIY*REPEAT)
SAIRY=SAIRY+(AIRY*REPEAT)
```
These statements accumulate test and repair manhours, respectively, for all LRUs through a General Support facility.

27. GSUY=ETI*SAIY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at the General Support facility. To post this work demand, ETI=1 is input, otherwise, ETI=0 is input.

28.
```
GSU=(EVIT*DI*168.*(GSUY/D(WI)))+((1.-EVIT)*DI*
        AINT((168.*(GSUY/D(WI)))+ZFL))
```
This statement computes the expected value of the service demand for Type I test equipment at all General Support facilities.

29.
```
GSUM=(EVIM*DI*168.*(GSUY/D(WIM)))+((1.-EVIM)*DI*
        AINT((168.*(GSUY/D(WIM)))+ZFL))
```
This statement computes the expected value demand on test manpower for LRUs through all General Support facilities.

30. GSURY=ETI*SAIRY
This statement controls posting out the accumulated repair work demand for Type I test equipment at the General Support service channel. ETI=1 is input for posting, otherwise ETI=0 is input.

31.
```
GSUR=(EVIR*DI*168.*(GSURY
        /D(WIR)))+((1.-EVIR)*DI*AINT((168.*
        (GSURY/D(WIR)))+ZFL))
```
This statement computes the expected value demand for repair manpower at all General Support facilities.

32. SADY=SADY+(ADY*REPEAT)
SADRY=SADRY+(ADRY*REPEAT)
These statements accumulate test and repair manhours, respectively, for all LRUs through one Depot facility.

33. AAIE=0.
IF(JTED.EQ.0.OR.JTED.EQ.2)AAIE=1.
These statements set the value for a qualifier (AAIE) that determines the type and location of test equipment. IF JTED=1 Type I test equipment is permitted at the Direct Support, General Support, and Depot sites. IF JTED=2 Type II test equipment is located at the Depot instead of Type I.

34. DEPY=ETI*(1.-AAIE)*SADY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at one Depot site. DEPY will have a value only when JTED=1.

35. DEP=(EVDT*168.*DD*(DEPY/D(WD)))+((1.-EVDT)*DD*
AINT((168.*(DEPY/D(WD)))+ZFL))
This statement computes the expected value of the service demand for Type I test equipment at all Depot facilities.

36. DEPM=(EVDM*168.*DD*(DEPY
/D(WDM)))+((1.-EVDM)*DD*AINT((168.*
(DEPY/D(WDM)))+ZFL))
This statement computes the expected value demand on test manpower for LRUs through all Depot facilities with Type I test equipment.

37. DEPRY=ETI*(1.-AAIE)*SADRY
This statement controls posting out the accumulated repair work demand for Type I test equipment at one Depot service channel. DEPRY will have a manpower value when JTED=1.

38. DEPR=(EVDR*168.*DD*(DEPRY
/D(WDR)))+((1.-EVDR)*DD*AINT((168.*
(DEPRY/D(WDR)))+ZFL))
This statement computes the expected value demand on repair manpower for LRUs through all Depot facilities with Type I test equipment.

39. DEPAIY=ETII*AAIE*SADY*(1.+FII)
This statement adds the Type II test equipment manhours for self-support to the test demand at one Depot service channel. DEPAIY will have a value only if the input flags ETII and JTED have the values 1 and 2, respectively.

40. DEPAIE=(EVDT*168.*DD*(DEPAIY
/D(WD)))+((1.-EVDT)*DD*AINT((168.*
(DEPAIY/D(WD)))+ZFL))

This statement computes the demand for Type II test equipment at all Depot sites.

41. DEPAIM=(EVDM*168.*DD*(DEPAIY
/D(WDM)))*((1.-EVDM)*DD*AINT((168.*
(DEPAIY/D(WDM)))+ZFL))
This statement computes the demand for test manpower of Type II test equipment at all Depot sites.

42. DEPARY=ETII*AAIE*SADRY
This statement controls posting out the accumulated repair work demand for Type II test equipment at the Depot.

43. DEPAR=(EVDR*168.*DD*(DEPARY
/D(WDR)))+((1.-EVDR)*DD*AINT((168.*
(DEPARY/D(WDR)))+ZFL))
This statement computes the demand for repair manpower at all Depot sites with Type II test equipment.

44. CAEY=CAEY+AEY*REPEAT
BCAEY=CAEY
These statements accumulate at one equipment level the manhour to test all LRUs in the system. This statement gives the same result as SAEY which is described in line 10 above.

45. CAERY=CAERY+AERY*REPEAT
BCAERY=CAERY
These statements accumulate at one Equipment Level the manhours to repair all LRUs in the system. This statement gives the same result as SAERY which is described in line 15 above.

46. CAOY=CAOY+AOY*REPEAT
CAORY=CAORY+AORY*REPEAT
These statements accumulate at one Direct Support level the manhours to test and repair, respectively, all LRUs in the system. These statements give the same result as SAOY and SAORY which are described in line 19 and 23 above.

47. CAIY=CAIY+AIY*REPEAT
CAIRY=CAIRY+AIRY*REPEAT
These statements accumulate at one General Support level the manhours to test and repair, respectively, all LRUs in the system.

48. CADY=CADY+ADY*REPEAT
CADRY=CADRY+ADRY*REPEAT
These statements accumulate at one Depot level the manhours to test and repair, respectively, all LRUs in the system. These statements give the same result as SAOY and SADRY which are described in line 32 above.

49.
```
HPD(1,1)=HPD(1,1)+24.*(1.-RF)
        *(1.+FE)*AME*REPEAT*TENMAN
HPD(1,2)=HPD(1,2)+(24.*RF*AME*REPEAT+SMF*24.
        *EE*REPEAT)*TENMAN
```
These statement accumulate the manhours per day to fault isolate and test, and to remove and replace, respectively, the LRUs at one equipment location.

50.
```
HPD(2,1)=HPD(2,1)+24.*(1.+FI)*AEY*REPEAT
HPD(2,2)=HPD(2,2)+24.*AERY*REPEAT
```
These statements accumulate the number of manhours per day to test and checkout, and to repair, respectively, the LRUs at one equipment location. The demand for Type I test equipment self-support (FI) is added to the test demand.

51.
```
HPD(3,1)=HPD(3,1)+24.*(1.+FI)*AOY*REPEAT
HPD(3,2)=HPD(3,2)+24.*AORY*REPEAT
```
These statements accumulate the number of manhours per day to test and repair, respectively, the LRUs through a Direct Support facility. The demand for Type I test equipment self-support (FI) is added to the test demand.

52.
```
HPD(4,1)=HPD(4,1)+24.(1.+FI)*AIY*REPEAT
HPD(4,2)=HPD(4,2)+24.*AIRY*REPEAT
```
These statements accumulate the number of manhours per day to test and repair, respectively, the LRUs through a General Support facility. The demand for Type I test equipment self-support (FI) is added to the test demand.

53.
```
HPD(5,1)=HPD(5,1)+24.*((1.+FI)*(1.-AAIE)+AAIE*
            *FII))*ADY*REPEAT
HPD(5,2)=HPD(5,2)+24.*ADRY*REPEAT
```
These statements accumulate the number of manhours per day to test and repair, respectively, the LRUs through a Depot facility. Manpower demand is in the unit of hours per calendar day. The demand for either Type I or Type II test equipment self-support is added to the test demand, depending on the value of AAIE.

N. This section of the program computes the development and procurement cost of installed equipment, stock, and test equipment. The personnel cost for test, repair and training as well as the salvage value of stock is also computed in this section. The salvage values in this program are all computed as negative numbers.

1.
```
AMPEAT = AMULT*REPEAT
```
This statement computes a multiplier which as a function of the number of identical LRUs in a materiel system, will convert costs to the output units specified by AMULT.

2. `CED=CEND*AMPEAT`
   This statement computes the development cost for all identical LRUs installed in a materiel system.

3. `CEP=EDEE*UCUP*SPE*AMPEAT`
   This statement computes the recurring costs for procurement of the installed LRUs at all Equipment installation. The factor SPE is used to adjust for prior expenditures ("sunk" costs).

4. `CEV=-SVE*EDEE*UCUP*AMPEAT`
   This statement computes the end of life salvage value of all installed LRUs.

5. `CEP=CEP+(CPE*AMPEAT)`
   This statement adds the nonrecurring costs for procurement of installed LRUs to the previously computed value for recurring costs.

6. `CET=CED+CEP+CEV`
   This statement computes the subtotal cost for all installed LRUs which includes development, procurement, and salvage value.

7. `CTSD=AMULT*(CI+CII+CCAL+CCSP+CV)`
   This statement converts the sum of the costs to develop test equipment into units specified by AMULT.

8. ```
   CTSP=((ESU+DSU+GSU+DEP)*CPI+(DEPAIE*CPII)+(EACAL*
           CALSET*CCALP)+(EACSP*CONTCT
           *CCSPP)+(ESI*CPV))*AMULT
   ```
   This statement computes the procurement costs for test equipment.

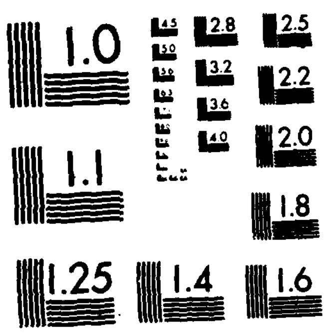
9. ```
   CMPRT=(168.*YR*AMULT)*((ETI*FI*((OD*CDMAN*TDMAN
           *(SAOY/D(WOM)))+(ED*CEMAN*TEMAN*(SAEY
           /D(WEM)))+(DI*CGMAN*TGMAN*(SAIY/
           D(WIM)))+((1.-AAIE)*TDPMI*CDPMAN*(SADY/
           D(WDM)))))+(ETII*FII*AAIE*TDPMII*CDPMAN
           *(SADY/D(WDM))))+ETE*FE*ED*TEMAN*
           CEMAN*SAME/WMR
   ```
   This statement computes the costs for personnel support of test equipment during the Operation and Support phase.

10. ```
    U2=(168.*YR*AMULT)*(ETI*FI
            *((1.-AAIE)*TDPMI*CDPMAN*(SADY/
            D(WDM)))+(ETII*FII*AAIE*TDPMII*
            CDPMAN*(SADY/D(WDM))))
    ```
    This statement computes the test personnel cost at the Depot level during the Operation and Support phase.
    ```

AD-A136 802 LOGAM (LOGISTIC ANALYSIS MODEL) VOLUME 3 TECHNICAL/PROGRAMMER MANUAL(U) COCKERHAM (JOHN M) AND ASSOCIATES INC HUNTSVILLE AL AUG 82

2/3

UNCLASSIFIED USAMICOM-DA-SR-D-82-2-VOL-3 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

11.
```
CTSR=((((ESU+DSU+GSU+DEP)*CRI)+(DEPAIE*CRII)
        +(EACAL*CALSET*CCALR)+(EACSP*CONTCT*
        CCSPR)+ESI*CRV)*YR*AMULT)+CMPRT
```
This statement computes the cost of materials and personnel to support test stations during the Operations and Support phase.

12.
```
CTSOFT=(CLRUPG+(P*CMODPG)+(CPUBII+CALPUB+
        CTCPUB+CPUBV))*AMPEAT
```
This statement computes the cost of development and programming of software for automated test equipment.

13.
```
CTSV=-CTSP*SVT
```
This statement multiplies the fraction of salvage cost with the test equipment procurement cost to get the salvage value of test equipment at the end of program.

14.
```
IF(IBG.EQ.1)WRITE(6,BUG8)
```
This statement prints the variables of NAMELIST/BUG8/ when debugging of a data case is essential.

15.
```
CTST=CTSD+CTSP+CTSR+CTSOFT+CTSV
```
This statement computes the cost associated with the development, procurement, maintenance, and software of test equipment (excluding salvage).

16.
```
YR12=12.*YR
```
This statement computes the number of months in the Operation and Support phase.

17.
```
CFR=(CFTD*((DEP*FTI)+(DEPAIE*FTII)))*AMULT*YR12
CFT=CFR
```
These statements compute the cost during the Operation and Support phase to house Type I and Type II test equipment at the Depot level.

18.
```
CMANE=CUCE*SMF*YR*ED*EE*REPEAT*AMULT
```
This statement computes the cost of scheduled maintenance for test crews at the Equipment level during the Operation and Support phase.

19.
```
CMPPY=((ETI*CTRI)+(ETII*CTRII)+(EACAL*CTRCAL)+
        (EACSP*CTRSPT)+ETE*CTRV)*AMULT
```
This statement computes the nonrecurring cost to set up training courses for all test equipment types.

20.
```
TRNG=AMULT*CTRA*ARA*YR*((TEMAN*ESUM)+
        (TGMAN*GSUM)+(TDMAN*DSUM)+(TDPMI*
        DEPM)+(TDPMII*DEPAIM)+(EACAL*TALMAN*
        CALSET)+(EACSP*TONMAN*CONTCT)+(TENMAN*
        (1.-RF)*DSIM))
```
This statement computes the recurring cost to train field

maintenance personnel for Equipment, General Support, Direct Support, Depot (Type I or II), Calibration, Contact, and Type I test equipment.

21.
```
CMPR=(((DSUM*CDMAN*TDMAN)+(GSUM*CGMAN)+
          TGMAN)+(CDPMAN*((DEPM*TDPMI)+
          (DEPAIM*TDPMII)))+(EACAL*
          CALSET*CALMAN*TALMAN)+(ESUM*
          CEMAN*TEMAN)+((1.-RF)*DSIM*
          CEMAN*TENMAN)+(EACSP*CONTCT*
          CONMAN*TONMAN))*YR*AMULT-CMPRT+
          TRNG+CMANE
```
This statement computes the cost of personnel at all test equipment facilities during the Operation and Support phase. Test equipment self-support cost (CMPRT) is excluded here since this cost has been included in the DSUM, GSUM, ESUM evaluations.

22.
```
U1=CDPMAN*YR*AMULT*(DEPM*TDPMI+DEPAIM*TDPMII)
```
This statement computes the cost of personnel support for Type I or Type II test equipment at Depot facilities during the Operation and Support phase.

23.
```
TRNGR=AMULT*CTRA*ARA*YR*(ESUR*TERMAN+
          DSUR*TDRMAN+GSUR*TGRMAN+DEPR*TDPRI+
          DEPAR*TDPRII+TENMAN*RF*DSIM)
```
This statement computes the cost to train repair personnel during the Operation and Support phase.

24.
```
CMPRR=YR*AMULT*((DSUR*CDRMAN*TDRMAN)+(GSUR*
          CGRMAN*TGRMAN)+(ESUR*TERMAN*CERMAN)+(RF*
          DSIM*CEMAN*TENMAN)+(DEPR*CDPRMN*TDPRI)+
          (DEPAR*CDPRMN*TDPRII))+TRNGR
```
This statement computes the cost of repair personnel of all facilities including the training cost (TRNGR) during the Operation and Support phase.

25.
```
U3=YR*AMULT*(DEPR*CDPRMAN*TDPRI+DEPAR*
          CDPRMN*TDPRII)
```
This statement computes the cost of repair personnel at the Depot for Type I or Type II test crews during the Operation and Support phase.

26.
```
CLS=FLM*(EACAL*CALSET*CALMAN*TALMAN+EACSP*CONTCT
          *CONMAN*TONMAN)*YR*AMULT
```
This statement computes the cost of civilian maintenance labor to support calibration/Type III and contact support/Type IV test equipment during the Operation and Support phase.

27.
```
PA=CMPR-U1+CMPRT-U2+CMPRR-U3-TRNG-TRNGR
```
This statement computes the pay and allowance for military

personnel by subtracting depot and training personnel costs from the total costs.

28. CMPT=CMPPY+CMPR+CMPRR
This statement computes the manpower cost for test and repair personnel, including training, during Operation and Support phase.

29. CIVP=SPEV*AMPEAT*((UCUP*QT)+(P*UCMP*QTM)+
(PP*FNSP*UCPP*QTP))
This statement computes the cost of inventory which includes cost for stocking LRUs (QT), modules (QTM), and parts (QTP) at all supply facilities. The factor SPEV is used to adjust for prior expenditures ("sunk" costs).

30. CIVR=AMPEAT*((YR8*QUA*FMWO*CKIT)
+(SPEVR*((UCUR*QC)+(P*UCMR*QCM)+
(PP*FNSP*UCPR*QCP))))
This statement computes the cost of consumed material during the Operation and Support phase. The costs include consumed LRUs (QC), modules (QCM), and parts (QCP).

31. CSVR=-SVR*CIVR
This statement computes the salvage value of consumed materials (CIVR).

32. RU=AMAX1(0.,QT-QQC)
RM=AMAX1(0.,QTM-QQCM)
RP=AMAX1(0.,QTP-QQCP)
These statements compute the difference (salvage value) in initial inventory stock and stock scrapped (consumed) during O&M phase. The greater of either the difference or zero is stored for LRUs, modules, and parts, respectively.

33. CRUT=UCUP*RU*AMPEAT
CRMT=P*UCMP*RM*AMPEAT
CRPT=PP*FNSP*UCPP*RP*AMPEAT
These statements compute the value of unconsumed stock at the end of O&M for LRUs, modules, and parts, respectively.

34. CIVV=-SVV*(CRUT+CRMT+CRPT)
This statement multiplies the salvage fraction (SVV) with the unconsumed stock values to determine the salvage value for stock left over at end of program.

35.
```
     IF(RU.LE.0.)GO TO 5075
     GO TO 5076
5075 RUD=0
     RUF=0
     GO TO 5077
5076 CONTINUE
     RUD=AMAXI(0.,QTD-QQCD)
```

```
     RUF=AMAXI(0.,U19-QQCF)
     IF(RUF.LE.0.)RUD=RU
     IF(RUD.LE.0.)RUF=RU
```

These statements compute the quantity of initial LRU stock salvaged at the Depot (RUD) and Field (RUF) at the end of the O&M phase. If there is not any salvage stock computed at these locations then the salvage stock for total system is used.

36.
```
5077 CONTINUE
     IF(RM.LE.0.)GO TO 5078
     GO TO 5079
5078 RMD=0.
     RMF=0.
     GO TO 5080
5079 CONTINUE
     RMD=AMAXI(0.,QTMD-QQCMD)
     RMF=AMAXI(0.,U8-QQCMF)
     IF(RMF.LE.0.)RMD=RM
     IF(RMD.LE.0.)RMF=RM
```

These statements compute the quantity of initial module stock salvage at the Depot (RMD) and Field (RMF) at the end of the O&M phase. The total salvage stock (RM) is used for RMF and RFD when there is no salvage stock computed at these levels.

37.
```
5080 CONTINUE
     IF(RP.LE.0.)GO TO 5081
     GO TO 5082
5081 RPD=0.
     RPF=0.
     GO TO 5083
5082 CONTINUE
     RPD=AMAX1(0.,QTPD-QQCPD)
     RPF=AMAX1(0.,U9-QQCPF)
5083 CONTINUE
     IF(RPF.LE.0.)RPD=RP
     IF(RPD.LE.0.)RFP=RP
```

These statements compute the quantity of initial part stock salvaged at the Depot (RPD) and Field (RPF) at the end of the O&M phase. The total salvage stock (RP) is used for RPF and RPD when there is no salvage stock computed at these levels.

38.
```
CRUTF=UCUP*RUF*AMPEAT
CRUTD=UCUP*RUD*AMPEAT
```

These statements compute the original cost of LRU stock salvaged at the field facilities and the Depot facilities, respectively.

39.
```
CRMTF=P*UCMP*RMF*AMPEAT
CRMTD=P*UCMP*RMD*AMPEAT
```

These statements compute the originial cost of module stock salvaged at the Field facilities and the Depot facilities, respectively.

40. CRPTF=PP*FNSP*UCPP*RPF*AMPEAT
CRPTD=PP*FNSP*UCPP*RPD*AMPEAT
These statements compute the originial cost of parts stock salvaged at the Field and Depot facilities, respectively.

41. CIVVD=-SVV*(CRUTD+CRMTD+CRPTD)
This statement uses the salvage fraction SVV to compute the value of all stock salvaged at the Depot facilities when the O&M phase is completed.

42. CIVVF=CIVV-CIVVD
This statement computes the salvage value of all stock at the Field facilities (Equipment, Direct Support, General Support.)

43. U12=TRNG+TRNGR
This statement computes the cost for training test and repair personnel during O&M.

44. U10=YR8*(QUA-QTD)*FMWO*CKIT*AMPEAT
This statement computes the O&M cost of modification kits for initial LRU buys at the Field facilities.(excluding Depot).

45. U11=UR8*QUA*FMWO*CKIT*AMPEAT-U10
This statement computes the O&M cost of modification kits for all initial LRU buys (QUA) and then subtracts U10 to get the cost at Depot only.

46. DMM=U10+U11-SVR*(U10+U11)
This statement computes the mod kit costs for all initial LRU buys with credit for the salvage value.

47. CIVRD=AMPEAT*((SPEVR*((UCUR*QCD)+(P*UCMR*QCMD)+
(PP*FNSP*UCPR*QCPD))))
This statement computes the cost of LRUs, modules and parts that are consumed over and above the quantity of materials originially stocked at the Depot levels. The factor SPEVR is used to adjust for prior expenditures ("sunk" cost).

48. CIVRF=CIVR-U11-CIVRD-U10
This statement computes the cost of LRUs, modules and parts that are consumed over and above the quantity of materials originally stocked at the Field facilities. This cost is reduced by the cost for mod kits at the field (U10) and Depot (U11).

49. CSVRF=-SVR*CIVRF

CSVRD=-SVR*CIVRD
These statements compute the salvage value for consumable material at the Field and Depot facilities, respectively. The salvage values here are for only the materials consumed over and above the original stocked quantities. SVR is the salvage factor input by the User.

50. CIVPF=SPEV*AMULT*REPEAT*((CUP*U19)+(P*
CMP*U8)+(PP*FNSP*CPP*U9))
This statement computes the cost of initial provisioning for LRUs (U19), modules (U8), and parts (U9) at the Field facilities. SPEV is the factor to adjust for prior expenditures ("sunk" costs).

51. CIVPD=SPEV*AMULT*REPEAT*((CUP*QTD)+
(PXCMP*QTMD)+(PP*FNSP*CPP*QTPD))
This statement computes the cost of initial provisioning for LRUs (QTD), modules (QTMD) and parts (QTPD) at the Depot facilities. SPEV is the factor to adjust for prior expenditures ("sunk" costs).

52. U21=(DEP*CRI+DEPAIE*CRII)*YR*AMULT
U20=CTSR-U21-CMPRT
These statements compute the cost of materials to support Type I and Type II test stations at the Depot and Field facilities, respectively.

53. REPSF=CIVRF+CSVRF+CIVVF+U20
RFPSD=CIVRD+CSVRD+CIVVD+U21
These statements compute the cost of consumed and support material after taking credit for the salvage value at the Field and Depot facilities, respectively.

54. DML=U1+U2+U3
This statement sums the personnel costs for support, test and repair at the Depot facilities.

55. CIVT=CIVP+CIVR+CSVR+CIVV
This statement computes the cost of initial provisions and consumed material after taking into account the salvage value (CSVR,CIVV).

56. CROR=SPEVR*AMPEAT*((CRU*(QC/D(QB)))+(P*CRM*
(QCM/D(QBM)))+(PP*FNSP*CRP*(QCP/D(QBP))))
CROT=CROR
These statements compute the cost of reordering LRUs, modules, and parts. Prior expenditures ("sunk" costs) are accounted for with the SPEVR factor.

57. CWHR=AMPEAT*YR12*((CSDSU*((CUBEU*QTO)+(P*
CUBEM*QTMO)+ (PP*FNSP*CUBEP*QTPO)))+(CSGSU*
((CUBEU*QTI)+ (P*CUBEM*QTMI)+ (PP*FNSP*

```
CUBEP*QTPI)))+(CSDEP*((CUBEU*QTD)+
(P*CUBEM*QTMD)+ (PP*FNSP*CUBEP*QTPD)))+
(CSESU*((CUBEU*QTE)+(P*CUBEM*QTME))))
CWHT=CHWR
```

These statements compute the cost of storage for LRUs, modules and parts at the Direct, General and Depot facilities.

58. CSMULT=(1.+(P*AQM)+(PP*FNSP*AQP))*AMPEAT

This statement computes a factor that will be used to compute the cost of entering and retaining line items in the supply system. This factor is the sum of the identical LRUs in a material system, the number of module types in the LRUs, and the number of non-standard part types in the LRUs. This factor is pertinent for supply to only one deployed material system.

59. CSAP=CSMULT*CEN

CSAR=CSMULT*CAD*YR

These statements compute the costs to enter a line item into the supply system and to retain the item in the supply system, respectively. A line item is made up of identical LRUs (REPEAT), module types (P*REPEAT) within the LRUs and the nonstandard part types (PP*FNSP*REPEAT) within the LRUs.

60. FSAC=(EDS*(AB(QTE)+AB(QTME)*P)+ODS*(AB(QTO)+

```
AB(QTMO)*P+ AB(QTPO)*PP*FNSP)+DIS*
(AB(QTI)+AB(QTMI)*P+AB(QTMD)*PP*FNSP))*
FSA*AMAX1(0.,YR-1.)*AMPEAT
```

This statement computes the cost of supply administration at the Field facilities per line item type. FSA is the cost factor per line item per supply point.

61. CSAR=CSAR+FSAC

CSAT=CSAP+CSAR

These statements compute the costs to retain and administer, and to enter and maintain line items in supply system, respectively.

62. U17=CROR+CWHT+CSAT

This statement computes the subtotal cost for reordering, storing, and administering a line item within the supply system.

63. CSHR=AMPEAT*((YR8*QUA*WTKIT*SHKIT)+(ONTIME

```
*((WU*SHU)+(WM+SHM)+(WP*SHP))))+AMPEAT*
CDIST*(WU*(QT+EDEE)+WM*QTM+WP*QTP)
CSHT=CSHR
```

These statements compute the total cost of shipping mod kits, LRUs, modules and parts for stocked and deployed items.

64.
```
CSHTF=AMPEAT*((YR8*U4*WTKIT*SHKITF)+(ONTIME*
        ((WU*SHUF)+(WM*SHMF)+(WP*SHPF))))+
        AMPEAT*CDIST*(WU*U4+WM*U8+WP*U9)
CSHTD=CSHT-CSHTF
```
These statements compute the cost of shipping mod kits, LRUs, modules and parts to the Field and Depot facilities, respectively.

65.
```
      X1=U20+U21+CMPRT-CTSR
      X2=CSVRF+CSVRD-SVR*(U10+U11)-CSVR
      X3=CSVRF+CSVRD-CSVR
      X4=CIVPF+CIVPD-CIVP
      X5=CIVVF+CIVVD-CIVV
      X6=PA+U1-CMPRT+U2-CMPR*U3+TRNG+TRNGR-CMPR
      X7=U10+U11+CIVRF+CIVRD-CIVR
      IPAS=IPAS+1
      WRITE(6,9756)IPAS,X1,X2,X3,X4,X5,X6,X7
9576  FORMAT(1H*AT CHECK*I5,7E16.8)
```
This set of codes is used as a debugging procedure in determining the correctness of computed costs. For the costs all to be correct then each value of X_i must be equal zero.

O. This section of the program computes the lifecycle phase costs, expected manpower costs, and cost growth factors for the lifecycle phases.

1.
```
CD=CED+CTSD+CTSOFT
CP=CEP+CTSP+CIVP+CSAP+CMPPY
CR=CTSR+CFR+CMPR+CMPRR+CIVR
        +CROR+CWHR+CSAR+CSHR+CSVR
CS=CEV+CTSV+CIVV
GCT=CD+CP+CR+CS
```
These statements sum individual cost areas for Development cost, Acquisition costs, O&M costs, end of program salvage value, and system total cost, respectively.

2.
```
PVCD=CD
PVCP=CP
PVCR=CR
PVCS=CS
```
These statements set the present value of each spending phase to the value just computed. The assumption is made that the computed values are the "constant dollar" values.

3.
```
DELTA=YR*AMULT*((CDMAN*TDMAN*(DSUM-
        (168.*OD*(DSUY/D(WOM)))))+(CDRMAN
        *TDRMAN*(DSUR-(168.*OD*(DSUY/D
        (WOR)))))+(CGMAN*TGMAN*(GSUM-(168.*
```

```
DI*(GSURY/D(WIM)))))+(CGRMAN*TGRMAN*
(GSUR-(168.*DI*(GSURY/D(WIR)))))*
(CEMAN*TEMAN*(ESUM-(168.*ED*(ESUY/
D(WEM)))))+(CEMAN*TERMAN*(ESUR-(168.*
ED*ESURY/D(WER))))+(DSIM-(168.*ED*ESME/
D(WMR))))
```

This statement computes the difference between the expected value and the dedicated manpower costs. If dedicated manpower values (DSUM, DSUR, GSUM, GSUR, ESUM, ESUR, DSIM) are computed as expected value then DELTA will be zero; otherwise a cost penalty of dedicated manpower as opposed to shared manpower will exist.

4.
```
ECMPT=CMPT-DELTA
EPCGT=GCT-DELTA
```
These statements compute expected value costs by subtracting the manpower cost penalty (DELTA) from the dedicated costs for maintenance manpower and system total, respectively.

5.
```
PDELTA=DELTA
```
This statement sets the difference between expected and dedicated manpower costs (DELTA) as the present value manpower cost penalty (PDELTA).

6.
```
    IF(FINT)400,40,400
400 CONTINUE
```
These statements test the value of the input variable FINT for the presence of a yearly interest rate. FINT is the net rate between discount rate and inflation rate. If FINT is non-zero then cost growth (yearly) factors are computed below; otherwise the logic to be executed will begin at statement 40.

7.
```
PART=1.+FINT
PART1=PART**YD
PART2=PART**YP
PART3=PART**(-YR)
PART4=PART**YZ
```
These statements compute cost growth factors for Annual Rate, Development phase, Production phase, O&M phase, and a Time Shifting phase, respectively. PART4 is used in time phasing the costs by shifting the point of reference at which present value is started.

8.
```
PVCD=(CD/D(YD))*((PART1-1)/FINT)*PART4*PART2
PVCP=(CP/D(YP))*((PART2-1.)/FINT)*PART4
FACTOR=((1.-PART3)/FINT*PART4
PVCR=(CR/D(YR))*FACTOR
PDELTA=(DELTA/D(YR))*FACTOR
PVCS=CS*PART3*PART4
```
These statements recompute the present value costs as a function of cost growth for Development (PVCD), Acquisition

(PVCP), O&M (PVCR), present value DELTA (PDELTA), and end of life salvage credit (PVCS). FACTOR is an internal computation used in the determination of PVCR and PDELTA.

9.
```
40 PVGCT=PVCD+PVCP+PVCR+PVCS
   EPVGCT=PVGCT-PDELTA
```
These statements compute the costs for present and expected present value totals, respectively. When FINT is input as zero, transfer is made to statement 40 from the IF test in line item 6 above, the totals would not include cost growth.

P. This section of the program accumulates individual costs from each LRU case to keep a running total of all LRUs evaluated in a case total. The cost of maintenance and initial provisioning are also included.

1.
```
CCET=CCET+CET
CCTS=CCTS+CTST
CCF=CCF+CFT
CCM=CCM+CMPT
CIV=CIV+CIVT
CRT=CRT+CROT
CWH=CWH+CWHT
CSA=CSA+CSAT
CSH=CSH+CSHT
CGT=CGT+GCT
```
These statements sum the costs for development of prime equipment (CET), development of test equipment (CTST), housing of test equipment at Depot (CFT), manpower to operate test equipment (CMPT), acquisition of prime equipment (CIVT), reordering of prime equipment (CROT), cost of storage for prime equipment (CWHT), entering and maintaining line items in the supply system (CSAT), shipping of prime equipment and mod kits (CSHT), and case total (GCT).

2.
```
F1=F1+PA
F2=F2+REPSF
F3=F3+DML
F4=F4+RFPSD
F5=F5+CSHTD
F6=F6+DMM
F7=F7+CLS
F8=F8+U12
F9=F9+CSHTF+CFT
F10=F10+U17
F11=F11+CED+CTSD
F12=F12+CEP+CEV+CTSV
F13=F13+CTSOFT
F14=F14+CMPPY
```

F15=F15+CIVP
F16=F16+CTSP
These statements accumulate case totals for the following maintenance support costs; pay and allowance for military personnel (PA), consumed and support material at field facilities (REPSF), personnel cost at Depot (DML), consumed and support material at Depot (RFPSD), shipping prime equipment and mod kits to Depot (CSHTD), mod kits (DMM) civilian maintenance labor (CLS), training test and repair personnel (U12), housing test equipment and shipping mod kits to field (CSHTF+CFT), maintaining line items in the supply system (U17), development of prime and test equipment (CED+CTST), procurement of installed LRUs less salvage value (CEP+CEV+CTSV), development of test software (CTSOFT training courses for test equipment (CMPPY), stock invent- / (CIVP), and procurement of test equipment (CTSP).

3. PCD=PCD+PVCD
PCP=PCP+PVCP
PCR=PCR+PVCR
PCS=PCS+PVCS
These statements accumulate the present value costs for Development, Acquisition, O&M, and Salvage, respectively.

4. PCGT=PCGT+PVCGT
This statement accumulates the present value cost total.

5. SDEL=SDEL+DELTA
This statement accumulates the difference (DELTA) between the expected value and the dedicated value of manpower costs.

6. SEPV=SEPV+EPVCGT
SEMPT=SEMPT+ECMPT
SEPC=SEPC+EPCGT
These statements accumulate the costs for the present value grand total, test equipment manpower, and the expected grand total, respectively. Total manpower cost is the personnel support for both test and prime equipment. An expected value is determined by subtracting DELTA from the dedicated value.

7. SPDEL=SPDEL+PDELTA
This statement accumulates the present value difference in dedicated manpower cost and expected manpower cost.

8. SPCR=PCR-SPDEL
This statement reduces the present value cost of the O&M phase by the accumulated manpower difference (SPDEL).

9. CCTSR=CCTSR+CTSR
This statement accumulates the support cost for all test

equipment. This includes both material and personnel cost.

10.
```
CTRF=CTRF+AMULT*CTRA*ARA*YR
        *((TEMAN*ESUM)+(TDMAN*DSUM)+
        (TGMAN*GSUM)+(EACAL*TALMAN*
        CALSET)*(EACSP*TONMAN*CONTCT)
        +(ESUR*TERMAN)+(DSUR*TDRMAN)+
        (GSUR*TGRMAN)+(DSIM*TENMAN))
CTRDEP=CTRDEP+((AMULT*CTRAD*ARAD*YR)*
        ((TDPMI*DEPM)+(TDPMII*DEPAIM)+
        DEPR*TDPRI+DEPAR*TDPRII))
```
These statements accumulate the O&M cost to train civilian personnel for test and repair at Field and Depot maintenance, respectively.

11.
```
CCMF=CCMF+((AMULT*YR)*(((DSUM
        *CDMAN*TDMAN)+(GSUM*CGMAN*
        TGMAN)+(EACAL*CALSET*CALMAN*
        TALMAN)+(EACSP*CONTCT*CONMAN*
        TONMAN)+(ESUM*CEMAN*TEMAN)+
        (ESUR*CERMAN*TERMAN)+(DSIM*
        CEMAN*TENMAN)+(DSUR*CDRMAN*
        TDRMAN)+(GSUR*CGRMAN*TGRMAN))-
        ((168.*ETI*FI)*((OD*CDMAN*TDMAN
        *(SAOY/D(WOM)))+(ED*CEMAN*
        TEMAN*(SAEY/D(WEM)))+(DI*CGMAN*
        TGMAN*(SAIY/D(WIM))))))-AMULT*
        YR*168.*ETE*FE*ED*CEMAN*TENMAN*
        (1.-RF)*SAVE/D(WRM)+CMANE

CCMD=CCMD+((AMULT*YR)*(((DEPM*CDPMAN*TDPMI)+
        (DEPAIM*CDPMAN*TDPMII)+(CDPRMN*
        ((DEPR*TDPRI)+(DEPAR*TDPRII))))-
        ((168.*CDPMAN*(SADY/D(WDM)))*
        ((ETI*FI*(1.-AAIE)*TDPMI)+(ETII*
        FII*AAIE*TDPMII)))))
```
These statements compute the cost of maintenance at the Field and Depot facilities, respectively. The costs computed up to the negative (-) sign in both CCMF and CCMD includes the cost of self-support for test equipment. Since the costs here are for maintenance only, the self-support costs are subtracted. CMANE, the cost of manpower at the equipment level is added to CCMF to provide the total field support personnel cost.

12.
```
CIVREC=CIVREC+CIVR
CSAREC=CSAREC+CSAR
```
These statements accumulate the cost of consumed material and the cost of inventory management, respectively.

13.
```
CQTE=UCUP*QTE*AMPEAT
CQTO=UCUP*QTO*AMPEAT
```

CQTI=UCUP*QTI*AMPEAT
CQTD=UCUP*QTD*AMPEAT
These statements compute the cost of initial LRU provisions at all Equipment, Direct Support, General Support, and Depot facilities, respectively. UCUP is the unit cost per LRU, the QTs are the quantities of LRUs stocked at all locations, and AMPEAT is the product of the cost conversion factor (AMULT) and the number of identical LRUs (REPEAT) per unique LRU in a material system.

14. CQTME=P*UCMP*QTME*AMPEAT
CQTMO=P*UCMP*QTMO*AMPEAT
CQTMI=P*UCMP*QTMI*AMPEAT
CQTMD=P*UCMP*QTMD*AMPEAT
These statements compute the cost of modules stocked at all Equipment, Direct Support, General Support, and Depot facilities, respectively. P is the number of module types per LRU, UCMP is the unit cost per module, the QTMs are the quantity of modules stocked of one type per LRU at all installations, and AMPEAT is the same as described in line item 13 above.

15. CQTPO=PP*FNSP*UCPP*QTPO*AMPEAT
CQTPI=PP*FNSP*UCPP*QTPI*AMPEAT
CQTPD=PP*FNSP*UCPP*QTPD*AMPEAT
These statements compute the cost of parts stocked at all Direct Support, General Support, and Depot facilities, respectively. PP is the number of part types per LRU, FNSP is the fraction of nonstandard parts, UCPP is the unit cost per part, the QTPs are the quanity of parts stocked of one type per LRU at all installations, and AMPEAT is described in line item 13 above.

16. CQTT=CQTO+CQTI+CQTD+CQTE
CQTU=CQTU+CQTT
These statements sum the initial provisions cost for LRUs stocked at all levels and accumulates (CQTU) these costs for each LRU case evaluated.

17. CQTMT=CQTMO+CQTMI+CQTMD+CQTME
CQTM=CQTM+CQTMT
These statements sum the initial provisions cost for Modules stocked at all levels and accumulates (CQTM) these costs for each LRU case evaluated.

18. CQTPT=CQTPO+CQTPI+CQTPD
CQTP=CQTP+CQTPT
These statements sum the initial provisions cost for Parts stocked at all levels and accumulates (CQTP) these costs for each LRU case evaluated.

Q. This section of the program is executed only if the output of summarized LRU costs are requested. The first statement checks the status of IFLAG which determines if the summarization logic is entered. LRU data are stored in array VV for each case, summed in array "C" for all LRUs in a data set and written to disk files (K001FX, K002FX) for output later.

1. IF (IFLAG.GT.0) GO TO 8027
   IFLAG is used to skip the output for LRUs summed separately. IFLAG greater than zero means to skip individual LRU case output; otherwise each LRU case is printed. The use of IFLAG in this statement is to skip the storage on disk (VV parameters) of LRU data and the summing of this data (C parameters) for intermediate print.

2. IF (SENSY.NE.0) GO TO 8710
   This statement checks for a non-zero value of the sensitivity analysis flag SENSY(1) and transfers to logic (statement 8710) that will set counters, position data file, and read the input data.

3. GO TO 8711
   This statement skips the sensitivity initialization section of code when SENSY=0 (sensitivity analysis is not requested).

4. ```
   8710 ICN = ICN + 1
        IF (ICN.GT.NDLRU) REWIND K009FX
        IF (ICN.GT.NDLRU)ICN = 1
        READ (K009FX) DM, UNS(1), UNS(2), UNS(3),
        UNS(4), UNS(5), DR
   ```
   These statements are used when sensitivity analyses are being performed. The LRU counter (ICN) is incremented until the last LRU in a data set (NDLRU) has been analysed. After the LRU data set is processed, the counter is reset (ICN=1). The LRU input data file is rewound and read for descriptive information (UNS) only. DM and DR are dummy parameters for storing the unused data read from K009FX.

5. 8711 CONTINUE
   This statement begins the logic for storing the output from an LRU data case into the VV array. The VV array is used to summarize data for sets of LRUs.

6. ```
   VV(1) = PVCGT
   VV(2) = PCGT
   VV(3) = CET
   VV(4) = CTST
   VV(5) = CFT
   ```

```
VV(6)  = CMFT
VV(7)  = CIVT
VV(8)  = CROT
VV(9)  = CWHT
VV(10) = CSAT
VV(11) = CSHT
VV(12) = GCT
VV(13) = QT
VV(14) = QTM
VV(15) = QTP
VV(16) = QUA
VV(17) = QMA
VV(18) = QPA
VV(19) = QC
VV(20) = QCM
VV(21) = QCP
VV(22) = RU
VV(23) = RM
VV(24) = RP
VV(25) = AOY
VV(26) = SAOY
VV(27) = AORY
VV(28) = SAORY
VV(29) = AIY
VV(30) = SAIY
VV(31) = AIRY
VV(32) = SAIRY
VV(33) = ADY
VV(34) = SADY
VV(35) = ADRY
VV(36) = SADRY
VV(37) = CAOY
VV(38) = CAORY
VV(39) = CAIY
VV(40) = CAIRY
VV(41) = CADY
VV(42) = DSU
VV(43) = DSU
VV(44) = DSUM
VV(45) = DSUR
VV(46) = GSU
VV(47) = GSUM
VV(48) = GSUR
VV(49) = DEP
VV(50) = DEPM
VV(51) = DEPR
VV(52) = DSUY
VV(53) = GSUY
VV(54) = DEPY
VV(55) = DEPAIE
VV(56) = DEPAIM
VV(57) = DEPAR
```

```
VV(58) = DEPAIY
VV(59) = EPVGCT
VV(60) = SEPV
VV(61) = ECMPT
VV(62) = DELTA
VV(63) = PDELTA
VV(64) = QTE
VV(65) = QTO
VV(66) = QTI
VV(67) = QTD
VV(68) = QTMO
VV(69) = QTMI
VV(70) = QTMD
VV(71) = QTPO
VV(72) = QTPI
VV(73) = QTPD
VV(74) = CQTE
VV(75) = CQTO
VV(76) = CQTI
VV(77) = CQTD
VV(78) = CQTT
VV(79) = CRUT
VV(80) = CQTMO
VV(81) = CQTMI
VV(82) = CQTMD
VV(83) = CQTMT
VV(84) = CRMT
VV(85) = CQTPO
VV(86) = CQTPI
VV(87) = CQTPD
VV(88) = CQTPT
VV(89) = CRPT
VV(90) = AEY
VV(91) = SAEY
VV(92) = AERY
VV(93) = SAERY
VV(94) = CAEY
VV(95) = CAERY
VV(96) = ESU
VV(97) = ESUM
VV(98) = ESUR
VV(99) = ESUY
VV(100) = QTME
VV(101) = CQTME
```

These statements store the results from an individual LRU data case into an array that will be written to a disk file for later retrieval when summarizing data cases. The descriptions of the variables stored into VV have been described in preceeding section of this document.

7. IF (ISET.EQ.1) GO TO 8503
This statement transfers logic to statement 8503 to read the

summarized LRU data array "C" from disk unit K001FX (UNIT 17) and to write the present LRU data array (VV) to disk unit K002FX (UNIT 18). Initially, ISET=0 and will remain zero until all the LRUs in a summarization set are processed, i.e., ICN=NDLRU.

8. IF (IATE.EQ.1) GO TO 8603
This statement transfers logic to statement 8603 to read the summarized LRU data array "C" from disk unit K002FX (UNIT 18) and to write the present LRU data to disk unit K001FX (UNIT 17). Initially, IATE=0 and will remain zero until all LRUs in a summarization set are processed, i.e., IATE=NDLRU.

9. WRITE (K001FX) (UNS(I),I=1,5)
WRITE (K001FX) (VV(I),I=1,101)
These statements write the LRU data case description and the data for the individual case to disk unit K001FX, respectively. This logic is entered only on the first NDLRU passes of the LRUs in a data concept. For the following summarization logic to work properly, the user must have the LRUs grouped in sets of NDLRU LRUs.

10. IF (ICN.EQ.NDLRU) GO TO 8501
When the first NDLRU set of LRUs are completed a transfer is made to statement 8501 to set a flag (ISET) that will prevent logic flow through the previous three statements.

11. GO TO 8502
The evaluation of the NDLRU set of LRUs in a data concept has not been completed, therefore, logic is transferred to statement 8502 which is outside the summarization logic.

12.
```
8501 ISET = 1
     ITAP = K001FX
     REWIND K001FX
     REWIND K002FX
     GO TO 8502
```
These statements are executed after the first set (NDLRU) of LRUs are completed. The data files are rewound for later use when LRU data will be read from the files, summarized, and written back to the files. ITAP is the file unit containing the data summarized from the previous set of LRUs. Transfer of logic is to statement 8502 to begin the evaluation of the next set of LRUs.

13.
```
8503 CONTINUE
     READ (K001FX) DDM1, DDM2, DDM3, DDM4, DDM5
     READ (K001FX) (C(I), I=1,101)
     DO 8508 I=1,101
8508 C(I)=C(I)+VV(I)
     WRITE (K002FX) DDM1, DDM2, DDM3, DDM4, DDM5
     WRITE (K002FX) (C(I),I=1,101)
```

```
IA8 = IA8+1
```

These statements are executed on even numbered sets of LRUs. LRU descriptive information (DDM1) and data summarized from previous LRU sets are read from K001FX. The summarized data array is added to the values for the present LRU case (VV) and written to disk unit K002FX. IA8 counts the number of passes completed through the even numbered set of LRUs.

14. IF (IA8-NDLRU) 8502, 8601, 8502

When all NDLRU LRUs in the even numbered data set have been processed logic transfer is made to statement 8601 to set flags for processing the next (odd) set of LRUs; otherwise, go to statement 8502 and complete the present set.

15.
```
8601 IA8 = 0
     IATE = 1
     ISET = 0
     ITAP = K002FX
     REWIND K001FX
     REWIND K002FX
     GO TO 8502
```

These statements are executed after all the LRUs (NDLRU) in an evened numbered set are processed. The even number set counter (IA8) is turned off and the odd number set flag (IATE) is turned on. The file number containing the latest summarization is stored in ITAP. The files are rewound and logic transfer is made to statement 8502 to begin processing of the next LRU set.

16.
```
8603 READ (K002FX) DDM1, DDM2, DDM3, DDM4, DDM5
     READ (K002FX) (C(I), I = 1,101)
     DO 8605 I = 1,101
8605 C(I) = C(I) + VV(I)
     WRITE (K001FX) DDM1, DDM2, DDM3, DDM4, DDM5
     WRITE (K001FX) (C(I), I=1,101)
     IA7 = IA7 + 1
```

These statements are executed on odd numbered sets of LRUs, except for the first set. LRU descriptive information (DDM1) and summarized data from previous sets are read from K002FX. The values in the summarized array (C) are added to the values for the present case (VV) and written to disk unit K001FX. IA7 counts the number of passes completed through the odd numbered set of LRUs.

17. IF (IA7 - NDLRU) 8502, 8609, 8502

When all NDLRU LRUs in the odd numbered data set have been processed, logic transfer is made to statement 8609 to set flags for processing the next (even) set of LRUs; otherwise, go to statement 8502 and complete the present set.

18.
```
8609 IA7 = 0
     IATE = 0
```

```
ISET = 1
ITAP = K001FX
REWIND K001FX
REWIND K002FX
```

These statements are executed after all the LRUs (NDLRU) in an odd numbered LRU data set are processed. The odd number set counter (IA7) and the odd number set flag (IATE) are turned off. The even numbered set flag (ISET) is turned on, ITAP set to the current summarized data file, and files rewound for next data set.

19. 8502 CONTINUE
    This statement is the terminal point after all LRU cases are summarized and written to disk file.

20. 8027 CONTINUE
    This statement is the transfer point when the option for requesting summarized output is not selected. When IFLAG=1 is input, the summarization process discussed above (section Q) will be shipped.

R. This section of the program prints the results for individual LRU cases. Each LRU output page includes the stock, work demands, and costs for the current LRU evaluated as well as cumulative values from the previous LRU evaluations. The output from this section can be inhibited by setting INHIB=1 in the NAMELIST /L/ data deck.

1. IF (INHIB.EQ.1.AND.NU.GE.0) GO TO 70
   This statement controls the print of an individual LRU output page. INHIB=1 inhibits the output of individual LRUs but INHIB will be overridden by NU, when NU is input less than zero. A negative value (-1, -2, -3) of NU allows the printout of case totals and grand totals and also allows the printout of individual LRU data even though INHIB=1.

2. CALL PAGE
   This statement calls subroutine PAGE to print the header information for the individual LRU output.

3. IF (SENSY(1).NE.0.) CALL SENSIT
   This statement calls subroutine SENSIT when a sensitivity case is being evaluated to print additional header information describing the nature of the sensitivity run.

4. WRITE (K004FX,3001)PVGCT,PCGT,COSTIS,AYZOS,AYZIS
   This statement prints the present value cost totals for each LRU case and for cumulative LRU case, the cost unit description, operational availability and inherent availability, respectively. The format for this WRITE statement as well as the other WRITE statements in this

section is not included.

5. WRITE (K004FX,3002)CET,CTST,CFT,CMPT,CIVT,CROT,CWHT,
CSAT,CSHT,GCT
This statement prints the cost values for prime equipment, test equipment, test equipment space, manpower, supply, ordering, storage, administration, shipping, and total, respectively.

6. WRITE (K004FX,3003)QT,QTM,QTP,QUA,QMA,QPA,QB,QBM,
QBP,QBP,QC,QCM,QCP,RU,RM,RP
This statement prints the quantity of LRUs, modules and parts for total provisions, initial provisions, reorder buys, consumables, and residuals, respectively.

7. WRITE (K004FX,3005) AOY,SAOY,AORY,SAORY,AIY,SAIY,
AIRY,SAIRY,ADY,SADY,ADRY,SADRY,CAOY,
CAORY,CAIY,CAIRY,CADY,CADRY
This statement prints the test and repair manhours for LRUs at Direct Support, General Support, and Depot, respectively. Each test or repair output includes the value from the present case (i.e., AOY) and the accumulated value of each case (i.e., SAOY). The last six variables in the list have the same values as SAOY, SAORY, SAIY, SAIRY, SADY, and SADRY.

8. IF (ETI.EQ.1) WRITE (6,3006) DSU,DSUM,DSUR,GSU,GSUM,
GSUR,DEP,DEPM,DEPR,DSUY,GSUY,DEPY
This statement prints the accumulated demands for Type I test equipment at the Direct Support, General Support, and Depot facilities, respectively. The order of output at each level is test equipment demand (i.e., DSU), test manpower demand (i.e., DSUM), and repair manpower demand (i.e., DSUR). The last three variables in the list are the accumulated value of test manhours at the three maintenance locations. This output is printed only if the posting of cumulative demands is requested by the input flag ETI.

9. IF(ETII.EQ.1) WRITE (6,3007) DEPAIE,DEPAIM,DEPAR,DEPAIY
This statement prints the accumulated demands for Type II test equipment at the Depot. The values printed are demand for test equipment, test manpower, repair manpower, and test manhours, respectively. This output is printed only if the posting of cummulative demands is requested by the input flag ETII.

10. WRITE (K004FX,1008)EPVGCT,SEPV,ECMPT,DELTA,PDELTA
This statement prints the expected value life cycle cost, cumulative expected value life cycle cost, expected test equipment manpower cost, delta between expected manpower cost and dedicated cost, and the present value delta, respectively.

11. WRITE (K004FX,3020)QTE,QTO,QTI,QTD,QTMO,QTMI,
QTMD,QTPO,QTPI,QTPD
This statement prints the value of the initial provisions fcr LRUs, modules, and parts at Equipment (LRUs only), Direct Support, General Support, and Depot.

12. WRITE (K004FX,3021)CQTE,CQTO,CQTI,CQTD,CQTT,CRUT,
CQTMO,CQTMI,CQTMD,CQTMT,CRMT,CQTPO,
CQTPI,CQTPD,CQTPT,CRPT
This statement prints the cost of initial provisions at Equipment (LRUs only), Direct Support, General Support, Depot, totals and residuals for LRUs, modules, and parts, respectively.

13. BETI=ETI
This statement sets the work demand flag at service channels for Type I test equipment into a temporary location BETI. BETI is transferred to subroutine SUPI through command block /SUPIN/.

14. PQTME=QTME
This statement sets the initial module provisions at Equipment into a temporary location PQTME. PQTME is transferred to subroutine SUPI through common block /SUPIN/.

15. ISUP = 1
IF (AB(GC+CI+CJ+GK).NE.0.)CALL SUPI (ISUP)
Subroutine SUPI is called anytime that maintenance policies at the Equipment level are in force. ISUP determines which Equipment level parameters are printed in SUPI.

S. This section of the program reinitializes program variables for the next LRU data case.

1. SAME=SAME*(1.-ETE)
This statement resets the accumulator for the hours to remove LRU per clock hour to zero when the work demand flag for Type V test equipment is on (ETE=1.).

2. SETD=((1.-ETI)*(1.-AAIG))+(AAIE*(1.-ETII))
SADY=SADY*SETD
SADRY=SADRY*SETD
These statements reset the accumulators for test and repair manhours at a Depot facility, unless the work demand flags (ETI,ETII) are off.

3. SAEY=SAEY*(1.-ETI)
BSAEY=SAEY
SAERY=SAERY*(1.-ETI)

BSAERY+SAERY
These statements reset the accumulators for test and repair manhours at an equipment location when the work demand flag for Type I test equipment is on (ETI=1).

4. SAIY=SAIY*(1.-ETI)
SAIRY=SIRY*(1.-ETI)
These statements reset the accumulators for test and repair manhours at a Direct Support location when the work demand flag for Type I test equipment is on (ETI=1.)

5. SAOY=SAOY*(1.-ETI)
SAORY=SAORY*(1.-ETI)
These statements reset the accumulators for test and repair manhours at a General Support location when the work demand flag for Type I test equipment is on (ETI=1.)

6. STI=ETI
STII=ETII
These statements save the values for the flags that control posting out of accumulated work demands at service channels for Type I and Type II test equipment, respectively. These flags are used when printing summarized outputs.

7. CPUBII=0.
CPUBV=0.
CALPUB=0.
CTCPUB=0.
These statements reset the non-recurring cost factors for technical data of Type II, Type V, Type III, and Type IV test equipment, respectively.

8. EACAL=0.
EACSP=0.
These statements reset the flags that control posting out of one time costs for Type III and Type IV test equipment and manpower.

9. CI=0.
CII=0.
CV=0.
CCAL=0.
CCSP=0.
These statements reset the non-recurring cost factors for developing Type I, Type II, Type V, Type III, and Type IV test equipment, respectively.

10. CEND=0.
CPE=0.
These statements reset the non-recurring cost factors for LRU development and production, respectively.

11. ITR=0.
This variable is in the NAMELIST /L/ input list, but is not used in the program.

12. IF(IBG.EQ.1) WRITE (6,BUG9)
This statement prints the list of variables from NAMELIST /BUG9/ when debugging of a data case is required. The input of IBG=1 will initiate this print.

13. IF(NU.GE.0) GO TO KAD (1,7,8,9)
This statement transfers logic to the areas of the program dictated by KAD unless the printout control flag NU is input as a negative value. A negative value for NU will send control to the next section of the program where case totals will be output.

T. This section of the program prints summarized outputs for distinct groups of LRUs and accumulates LRU case totals for later output as a grand total. The summarized outputs are for those groups of LRUs selected by the input parameter NDLRU. For instance, if NDLRU=4 the output for every fourth LRU is summed together (i.e., $LRU_1 + LRU_5$, $LRU_2 + LRU_6$, etc.). The LRUs must be input in NDLRU sets for this summarization process to work properly. The summarized data are printed after all individual LRU data cases are printed and prior to the grand total page. An input value for NU that is less than or equal to a -1 will initiate this output. The summarized data array (C) that is output here was discussed in section Q above.

1.

```
    DO 66 I=1,35
66 SUM(I) = SUM(I) + CUM(I)
```

These data statements accumulate case totals for later output as grand totals. The data for each individual LRU case is accumulated into CUM. When a cost total (a group of LRU cases) is completed, the totals can be printed and accumulated into SUM with other case totals. After all case totals are completed a grand total can be output from the values accumulated in SUM. To output the case totals, the variable NU=-1 is input with the last LRU case in the case. The flag IS=1 must be input also to reset the case total accumulator (CUM) for the next set of LRU cases. SUM (grand total) is printed whenever the value of NU is a -2 or -3. CUM is equivalenced to the first 35 variables in COMMON BLOCK /ZERO/. The evaluation of the CUM values were discussed in line items P-1, P-3 through P-12, and P-16 through P-18.

2.

```
    DO 584 I=1,16
584 SDAM(I) = SDAM(I) + SDA(I)
```

These statements accumulate grand totals for maintenance support costs and are output in the same manner as discussed

in the previous statement. SDA contains the case totals and it is equivalenced to the F1 through F16 variables that were discussed in line item P-2.

3. PRINT= 0.
This flag controls the output of the grand total and the resetting of the grand total accumulator. When NU= -2 or -3 and PRINT=0., the logic for printing the grand total is entered. The program sets IPRINT=1 to prevent entering the grand total print logic again on the current pass.

4. 69 CONTINUE
This is the return statement for printing summarized LRU data and grand totals. The output statements are the same for case totals and grand totals. The first pass through will print the case total, if grand totals are requested (NU= -2 or -3) a jump back to this statement is made to print the grand totals.

5. IPAGE=IPAGE-1
This statement sets the page counter to be printed at the top of each page of output. The page counter is a negative accumulation to achieve an output effect only. For example, page 17 would be printed as "-17-" where the first dash is the negative sign and the dash following the number is placed there with a Hollerith statement (1H-).

6. IF (PRINT.EQ.0.)WRITE (6,5041)IPAGE,TEXT,ANLYSIS,DATE
This statement prints the title page for a case total output.

7. IF (IFLAG.GT.0)GO TO 8011
This statement will skip the summarization output of related LRU cases when a positive non zero value of IFLAG is input. Otherwise the related LRU cases that were accumulated into the "C" array from the VV values will be output depending on the status of the next statement.

8. IF (PRINT-1.)8011,8010,8011
This statement inhibits the output of summarized data cases until all LRU cases have been evaluated. Until a value of -2 or -3 is input for NU the value of PRINT will remain zero; therefore sending control past the output statements to statement 8011.

9. 8010 REWIND ITAP
This statement readies the summarized LRU data file to begin reading information to print.

10. DO 8014 KKK=1,NDLRU
This statement is the beginning of the DO loop for reading the summarized data from file ITAP. NDLRU is the number of

related LRU cases for which a summarization was performed.

11. READ (ITAP)(UNS(I),I=1,5)
READ (ITAP)(C(I),I=1,101)
These statements read the unit descriptions and the summarized data array, respectively, from the current data file (ITAP).

12. WRITE(K004FX,8005)IPAGE,TEXT,UNS,ANLYIS,TLRU,DATE
This statement outputs a page header for the summarized data case.

13. WRITE (K004FX,8015) C(1),C(2),COSTIS
This statement prints the summarized present value total (PVCGT), present value grand total, (PCGT) and the cost unit description (COSTIS), respectively.

14. WRITE(K004FX,3002) (C(I),I=3,12)
This statement prints the summarized value for development of prime equipment (CET), development of test equipment (CTST), housing of test equipment at Depot (CFT), manpower to operate test equipment (CMPT), acquisition of prime equipment (CDVT), reordering of prime equipment (CROT), cost of storage for prime equipment (CWHT), entering and maintaining line items (CSAT), shipping (CSHT), and grand cost total, (CGT), respectively.

15. WRITE (K004FX,8016)(C(I),I=13,24)
This statement prints the summarized values for the initial provision quantities (QT,QTM,QTP), consumed quantities (QUA,QMA,QPA), and residual quantities (RU,RM,RP), respectively.

16. WRITE (K004FX,3005) (C(I),I=25,42)
This statement prints the summarized values for manhours and costs to test and repair LRUs at Direct, General, and Depot facilities. The variables output are the summarized values for AOY, SAOY, AORY, SAORY, AIY, SAIY, AIRY, AIRY, ADY,SADY, ADRY, SADRY, CAOY, CAORY, CAIY, CAIRY, CADY, and CADRY, respectively.

17. IF(STI.EQ.1.)WRITE (6,3006) (C(I),I=43,54)
This statement prints summarized values for the accumulated work demands of Type I test equipment at Direct Support, General Support and Depot. These outputs are printed only if the work demand flag for Type I test equipment was turned on (ETI=1.) in an individual LRU data case. The outputs here are summarized values for DSU, DSUM, DSUR, GSU, GSUM, GSUR, DEP, DEPM, DEPR, DSUY, GSUY, and DEPY, respectively.

18. IF (STII.EQ.1.) WRITE (6,3007) (C(I),I=55,58)
This statement prints summarized values for the accumulated

work demands of Type II test equipment at the Depot facilities. These outputs are printed only if the work demand flag for Type II test equipment was turned on (ETII=1.) in an individual LRU data case. The outputs here are the accumulated values for DEPAIE, DEPAIM, DEPAR, and DEPAIY, respectively.

19. WRITE (K004FX,1008) (C(I),I=59,63)
This statement prints summarized values for the expected value lifecycle cost (EPVGCT), cumulative expected lifecycle cost (SEPV), expected test equipment manpower cost (ECMPT), delta between expected and dedicated manpower cost (DELTA), and present value delta (PDELTA), respectively.

20. WRITE (K004FX,3020)(C(I),I=64,73)
This statement prints the summarized values for the initial provisions of LRUs (QTE, QTO, QTI, QTD), modules (QTMO, QTMI, QTMD), and parts (QTPO, QTPI, QTPD), respectively.

21. WRITE (K004FX,3021) (C(I),I=74,89)
This statement prints the summarized cost of initial provisions for LRUs (CQTE, CQTO, CQTI, CQTD, CQTT, CRUT), modules (CQTMO, CQTMI, CQTMD, CQTMT, CRMT), and parts (CQTPO, CQTIPI, CQTPD, CQTPT, CRPT), respectively. CQTT, CQTMT, and CQTPT are the total provisioning costs and CRUT, CRMT, and CRPT are the end of life salvage value for LRUs, modules, and parts, respectively.

22. ATEST=AB(C(90)+C(91)+C(92)+C(93)+C(94)+C(95)+C(96)+
C(97)+C(98)+C(99)+C(100)+C(101)
This statement uses the function AB to determine if there are work demands and costs at the organizational maintenance level for policies GC, GI, GJ, and GK. If any value in the argument list is non-zero, then ATEST=1; otherwise, ATEST is zero. The variables in the argument list are the summarized values for AEY, SAEY, AERY, SAERY, CAEY, CAERY, ESU, ESUM, ESUR, ESUY, QTME, and CQTME, respectively. When ATEST=1 subroutine SUPI is called to print the results.

23. IF(IBG.EQ.1.)WRITE (6,BUG10)
This statement prints the variable list in NAMELIST /BUG10/ when debugging of a data case is required.

24. IF(PRINT.EQ.1.)GO TO 889
GO TO 888
These statements are unnecessary since there is no way to get to this logic unless PRINT=1. The statement discussed in line item 8 above will bypass all the summarized output logic by transferring to statement number 8011 (line item 30) when PRINT is not equal to one.

25. 889 IF(ATEST.EQ.0.)GO TO 888

```
ISUP=2
IPAGE=IPAGE-1
WRITE (K004FX,8005)IPAGE,TEXT,UNS,ANLYIS,TLRU,DATE
CALL SUPI(ISUP)
```

These statements result in the output of summarized work demands and costs at the organizational level when maintenance policies GC, GI, GJ and GK are used. Subroutine SUPI prints the results and it is called when ATEST=1, which signifies that organizational level maintenance policies were used.

26.
```
888 IPAGE=IPAGE-1
```
This statement is the transfer point to bypass printing the organizational level results and also sets the page counter for the next page of output.

27.
```
8014 CONTINUE
```
This statement is the end of the DO Loop for printing summarized results. After NDLRU loops have been made through the preceeding statements, logic will begin with the next statement.

28.
```
     DO 8018 I=1,101
8018 C(I)=0.0
```
These statements reset the values in the summarized data array.

29.
```
ISET=0
REWIND K001FX
IATE=0
IA7=0
IA8=0
REWIND K002FX
IF(SENSY(1).NE.0)REWIND K009FX
```
These statements reset the variables and rewind the files used in summarizing LRU data cases. After this logic the program is ready to summarize results for a sensitivity analyses or another LRU data concept.

30.
```
8011 CONTINUE
```
This statement is the transfer point for bypassing summarized LRU outputs. The logic for reaching this point is discussed in line item 8 above.

U. This section of the program computes the manpower required to perform test, checkout and repair at all organizational, Direct Support, General Support, and Depot maintenance facilities. The manpower requirements computed here are the accumulated values for a case total. The case total and grand total outputs are also included in this section. The grand total output uses the same print statements as the case total. When grand total outputs are requested

the SUM array containing the grand totals is stored into the CUM array. Since the CUM array is equivalenced to the list of variables in the print statement, the grand totals can be output with another pass through the case total print statements.

1. CQTUMP=CQTU+CQTM+CQTP
   This statement computes the total provisioning cost for LRUs, modules and parts.

2. CCMFD=CCMF+CCMD
   This statement computes the total cost of maintenance manpower at the Field and Depot facilities.

3. CTR=CTRF+CTRDEP
   This statement computes the total cost of training civilian personnel at the Field and Depot maintenance facilities.

4. CTREC=CCTSR+CCF+CCMFD+CTR+CIVREC+CRT+CWH+CSAREC+CSH
   This statement computes the total recurring investment costs.

5. ```
   PERS(1,1)=HPD(1,1)*7./D(WMR)
   PERS(1,2)=HPD(1,2)*7./D(WMR)
   PERS(1,1)=ED*(ETEI*PERS(1,1)+(1.-ETEI)*AINT
               PERS(1,1)+ZFL))
   PERS(1,2)=ED*(EREI*PERS(1,2)+(1.-EREI)*AINT(
               PERS(1,2)+ZFL))
   ```
   These statements compute case totals for the number of men per week that is required to fault isolate and test (PERS(1,1)) and to remove and replace (PERS(1,2)) LRUs at all organizational maintenance locations. If the expected value flag ETEI is on (1), the expected value is computed for PERS; otherwise the value is rounded up according to the input value for ZFL. WMR is the number of hours in a work week for repair crews.

6. ```
   PERS(2,1)=HPD(2,1)*7./D(WEM)
   PERS(2,2)=HPD(2,2)*7./D(WER)
   ```
   These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs at one organizational maintenance location. WEM and WER are the scheduled work weeks in hours for test and repair equipment manpower at the organizational level.

7. ```
   PERS(3,1)=HPD(3,1)*7./D(WOM)
   PERS(3,2)=HPD(3,2)*7./D(WOR)
   ```
   These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs at one Direct Support maintenance location. WOM and WOR are the scheduled work week in hours for test and repair equipment manpower at the
   ```

Direct Support level.

8.
```
PERS(4,1)=HPD(4,1)*7./D(WIM)
PERS(4,2)=HPD(4,2)*7./D(WIR)
```
These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs at one General Support maintenance location. WIM and WIR are the scheduled work week in hours for test and repair equipment manpower at the General Support level.

9.
```
PERS(5,1)=HPD(5,1)*7./D(WDM)
PERS(5,2)=HPD(5,2)*7./D(WDR)
```
These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs at one Depot maintenance location. WDM and WDR are the scheduled work week in hours for test and repair equipment manpower at the Depot level.

10.
```
PERS(2,1)=TEMAN*ED*(EVEM*PERS(2,1)+(1.-EVEM)*
          AINT(PERS(2,1)+ZFL))
PERS(2,2)=TERMAN*ED*(EVER*PERS(2,2)+(1.-EVER)*
          AINT(PERS(2,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TEMAN, TERMAN) and by the number of maintenance locations (ED) to get the total number of men per week for test and repair, respectively, at all organizational levels. The expect value flags (EVEM,EVER) are used to get either the computed value or an integer value that is rounded according to the value of ZFL.

11.
```
PERS(3,1)=TDMAN*OD*(EVOM+PERS(3,1)*(1.-EVOM)*
          AINT(PERS(3,1)+ZFL))
PERS(3,2)=TDRMAN*OD*(EVOR*PERS(3,2)*(1.-EVOR)*
          AINT(PERS(3,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TDMAN, TDRMAN) and the by the number of Direct Support locations (OD) to get the number of men per week for test and repair, respectively, at all Direct Support locations. The expected value is an integer value that is rounded according to the value of ZFL.

12.
```
PERS(4,1)=TGMAN*DI*(EVIM*PERS(4,1)+(1.-EVIM)*
          AINT(PERS(4,1)+ZFL))
PERS(4,2)=TGRMAN*DI*(EVIR*PERS(4,1)+(1.-EVIR)*
          AINT(PERS(4,2)+ZFL))
```
These statement multiply the number of crews by the number of men per crew (TGMAN, TGRMAN) and by the number of General Support locations (DI) to get the number of men per week for test and repair, respectively, at all General Support locations. The expected value flags (EVIM, EVIR) are used to get either the computed value or an integer value that is

rounded according to the value of ZFL.

13.
```
PERS(5,1)=DD*((1.-AAIE)*TDPMI+AAIE*TDPMII)*(EVDM*
          PERS(5,1)+(1.-EVDM)*AINT(PERS(5,1)+ZFL))
PERS(5,2)=DD*((1.-AAIE)*TDPRI+AAIE*TDPRII)*(EVDR*
          PERS(5,2)+(1.-EVDR)*AINT(PERS(5,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TDPMI, TDPRI) and by the number of Depot locations (DD) to get the number of men per week for test and repair, respectively, at all Depot locations. The expected value flags (EVDM, EVDR) are used to get either the computed value or an integer value that is rounded according to ZFL. The value of AAIE determines if the test equipment is of Type I or Type II. The number of men per crew will be TDPMII and TDPRII for Type II equipment.

14.
```
WPD(1,1)=HPD(1,1)*ED
WPD(1,2)=HPD(1,2)*ED
```
These statements compute case totals for the manhours per day to fault isolate and test, and to remove and replace, respectively, the LRUs at all equipment installations.

15.
```
WPD(2,1)=HPD(2,1)*ED
WPD(2,2)=HPD(2,2)*ED
WPD(3,1)=HPD(3,1)*OD
WPD(3,2)=HPD(3,2)*OD
WPD(4,1)=HPD(4,1)*DI
WPD(4,2)=HPD(4,2)*DI
WPD(5,1)=HPD(5,1)*DD
WPD(5,2)=HPD(5,2)*DD
```
These statements compute case totals for the manhours per day to test and repair the LRUs at all Equipment, Direct Support, General Support, and Depot facilities, respectively.

16.
```
      DO 5058 I1=1,5
      DO 5058 I2=1,2
      WPD(I1,I2)=365.25*WPD(I1,I2)
      IF((I1.LE.2).AND.(I2.LE.2))GO TO 5059
      GO TO 5060
5059  WPDL(I1,I2)=WPD(I1,I2)/EDS
      PERL(I1,I2)=PERS(I1,I2)/EDS
      GO TO 5058
5060  IF((I1.EQ.3).AND.((I2.EQ.1).OR.(I2.EQ.2)))GO TO 5061
      GO TO 5062
5061  WPD(I1,I2)=WPD(I1,I2)
      PERL(I1,I2)=PERS(I1,I2)/OD
      GO TO 5058
5062  IF((I1.EQ.4).AND.((I2.EQ.1).OR.(I2.EQ.2)))GO TO 5063
      GO TO 5064
5063  WPDL(I1,I2)=WPD(I1,I2)/DI
      PERL(I1,I2)=PERS(I1,I2)/DI
```

```
     GO TO 5058
5064 IF((I1.EQ.5).AND.((I2.EQ.1).OR.(I2.EQ.2)))GO TO 5065
     GO TO 5058
5065 WPDL(I1,I2)=WPD(I1,I2)/DD
     PERL(I1,I2)=PERS(I1,I2)/DD
5058 CONTINUE
```

The statements in this nested Do Loop compute total manhours per year (WPD), manhours per year per installation (WPDL), and the number of men per week per installation (PERL) required to checkout, replace, test, and repair equipment. The statement at statement number 5061 is most likely in error. The statement most likely should be (WPDL(I1,I2)=WPD(I1,I2)/OD). The information computed here is used for case total outputs.

17.
```
WRITE(K004FX,2001)COSTIS,CCET,CCTS,CCTSR,
          CCF,CCF,CCM,CCMF,CCMD,CCMFD,CTRF,CTRDEP,
          CTR,CIV,CIVREC,CRT,CRT,CWH,CWH,CSA,CSAREC,
          CSH,CSH, CGT,CTREC,PCD,CQTU,PCP,CQTM,PCR,
          CQTP,PCS,CQTUMP,PCGT
WRITE (6,12)
```

These statements print the output for both a case total and the grand total. Anytime the value of NU is less than zero a case total will be printed. A value of -2 or -3 for NU will result in the grand total array SUM being stored into the case total array CUM and logic is transferred back through the above logic to give the grand total printout. CUM is equivalenced to CCET (second argument in list above) where CCET is also the first argument in the COMMON /ZERO/ list.

18.
```
J=3
```

This statement is a flag used in Subroutine EIGHT for determining which summary table to print. When subroutine EIGHT is called from the main program (LOGAM1), system maintenance support costs are output in a lifecycle phased-cost element format.

19.
```
UVA(1)=F1
UVA(2)=F2
UVA(3)=F3
UVA(4)=F4
UVA(5)=F5
UVA(6)=F6
UVA(7)=F7
UVA(8)=F8
UVA(9)=F9
UVA(10)=F10
UVA(11)=F11
UVA(12)=F12
UVA(13)=F13
UVA(14)=F14
UVA(15)=F15
```

UVA(16)=F16
These statements store the accumulated case total maintenance support costs into the array UVA for output in Subroutine EIGHT. The description for the variables can be found in line item P-2. UVA is passed to subroutine EIGHT with the COMMON /DAPAM/ statement.

20. IPAGE=IPAGE-1
This statement sets the page counter for the system maintenance support cost page.

21. CALL EIGHT (J)
This statement calls subroutine EIGHT to print the system maintenance support costs. This subroutine is called for both case total and grand total outputs. The significance of the variable J was discussed in line item U-18, and will be discussed further in the Subroutine EIGHT writeup.

22. IF(PRINT.EQ.1.)GO TO 77
This statement is used to by-pass the next section of code when the grand total flag (PRINT=1) is turned on. This next section of code outputs LRU system and subsystem operational availabilities, and manpower loadings for test and repair. This output is printed only after the case totals are printed.

23. WRITE (K004FX,1023)(CAYZ(I),I=1,NA)
WRITE (K004FX,1024)(CAYZI(I),I=1,NA)
These statements print operational and inherent availabilities, respectively. The first location of each array contains the system availability while the remaining NA locations contain LRU subsystem availabilities. This print will be found after a case total has been output.

24. WRITE (K004FX,4001) WPD,PERS
This statement prints the case totals at all maintenance locations for the manhours per year and the number of men per week, respectively.

25. WRITE (K004FX,5067)WPDL, PERL
This statement prints the case totals at one location per maintenance level for the manhours per year and the men per week, respectively.

26. 77 CONTINUE
This is the transfer statement to by-pass the print statements described in the last four statements.

27. IF(NU.EQ.-1)GO TO KAD(1,7,8,9)
IF(NU.EQ.-2.AND.PRINT.EQ.1.)GO TO KAD(1,7,8,9)
IF(NU.LE.-3.AND.PRINT.EQ.1.)GO TO 75
These statements will either cause a transfer of logic to

the section of code specified by KAD or load the grand total outputs for printing, or reset the grand total accumulators (GO TO 75). When the value of NU is negative the case total output will have been printed, therefore, NU=-1 will send the logic to one of the statements assigned to KAD. NU=-2 or -3 specifies a grand total output. The difference being that NU=-3 will reset the grand total accumulators after the grand total is printed. The flag PRINT=1 is set to indicate the grand totals were printed. When PRINT=0 and NU=-2, or -3, the grand total accumulators (SUM) will be stored in the case total accumulator (CUM) for grand total output. KAD=1 is the normal transfer to statement number 1 for the next individual LRU data case. KAD=7 or 8 transfers logic back into the sensitivity section of the program. Only when sensitivity analyses are being performed will KAD=7 or 8. KAD=9 is assigned after all sensitivity cases have been evaluated or an error was detected in sensitivity inputs. Statement 9 (KAD=9) is at the beginning of the program logic where accumulator arrays are reset. The setting of KAD in the sensitivity logic of this program is described in line items H-16, H-17, and H-18.

28. PRINT=1.
This statement is a flag to prevent entering this section of code once a grand total case is printed. It is used in the above statements.

29.
```
   DO 68 I=1,35
68 CUM(I)=SUM(I)
```
These statements store the grand total outputs SUM into the case total array CUM for printing. The grand total output used the same print statements as the case totals which is described in line item U-17 above. The grand totals are for those variables listed in COMMON /ZERO/.

30.
```
    DO 585 I=1,16
585 SDA(I)=SDAM(I)
```
These statements store the maintenance support grand total outputs SDAM into the case total array SDA for printing. The grand total output uses the same print subroutine (EIGHT) as did the case total output. These outputs are described in line items U-18 through U-21 above.

31. GO TO 69
This statement transfers logic back through the same code that was used to print case totals, but this time grand totals and summarized LRU data cases are printed. The summarized cases will be printed first. The description of this logic beginning with statement 69 is found in line item T-4.

32.
```
75 DO 67 I = 1,35
```

```
 67 SUM(I)=0.
    DO 586 I = 1,16
586 SDAM(I) = 0.
```

These statements reset the grand total accumulators. Only when NU=-3 will this logic be entered. SUM is the accumulator array for the grand totals printed in this routine while SDAM is the accumulator array for grand total maintenance support costs printed in subroutine EIGHT.

33.
```
GO TO KAD(1,7,8,9)
```

This statement transfers program logic to the appropriate section of code that is dictated by the value assigned to KAD.

34.
```
19 WRITE (K004FX,21) SENSY
21 FORMAT (1X,9HBAD SENSY/34(8E14.7/))
   GO TO 9
91 WRITE (K004FX,92)
92 FORMAT (1X,18HBAD SENSY SEQUENCE)
   GO TO 9
```

These statements print diagnostics when errors are encountered in sensitivity input data. After each diagnostic, a logic transfer is made to statement 9 to reinitialize accumulators and to read another data case. The transfer to either statement 19 or statement 91 is from the sensitivity section. A description of these transfers can be found in line items H-3 and H-6 above.

35.
```
3 STOP
  END
```

This statement terminates program execution. The only way to get to this logic is by reading on end-of-file on the input data file. An end-of-file occurs when all input data has been read. The transfer to statement 3 is described in line items F-10 and F-11 above. This isn't the only way to terminate execution of the program. An NU=-4 in a last data case will also terminate execution. This logic is described in line item F-21.

BLKDAT

This is a BLOCK DATA routine that initializes program variables. This routine causes the variables referenced here to assume the assigned values before the first executable statement is entered. This routine can not be called from another routine and the values are initialized only once per program start.

1. ```
   COMMON/SENS/SENSY(266),NRULE(12),NVAR(12),MODE,
           KPASS,NRU,LRU,NPASS
   COMMON/POUT/P(16)
   COMMON/T/T(2000)
   COMMON/OFF/O(6)
   COMMON/ENLM/E(7)
   COMMON/INPUT/SAVI(229),SAVA(45),SAVG(24),JSAV(5),
           SAVSA(22),SAV13(14)
   ```
   These statements are used to pass the variables initialized here to other routines in the program.

2. `DATA P/13*0.,1.,0.,0./`
   This statement initializes the printout array used in subroutine OPER. The 14th value which is initialized to 1 is for the AMULT variable. COMMON block/POUT/ transmits this data.

3. `DATA T/2000*0./`
   This statement initializes the TOE data array that is input in OPER with NAMELIST/TOE/. COMMON block /T/ transmits this data.

4. `DATA O/6*0./`
   This statement initializes the array for Officer personnel and pay and allowance. This data is transmitted to subroutine OPER and the routines referenced by OPER with COMMON block /OFF/.

5. `DATA E/7*0./`
   This statement initializes the enlisted personnel data array. This data is transmitted to subroutine OPER and the routines referenced by OPER with COMMON block /ELEM/.

6. `DATA SAVSA/19*0.,0,0,1./`
   This statement initializes values in COMMON block /INPUT/. The variables initialized are SIN(19),IIN(2), and TENMAN. Reference line item A-16 of LOGAM1 for a description of SIN and IIN.

7. ```
   DATA SAVI/0.,1.,73*0.,3*1.,5*0.,3*1.,0.,2*1.,0.,
           2*1.,0.,2*1.,4*0.,3*1.,2*0.,1.,6*0.,4*1.,
           5*0.,1.,5*0.,3*1.,12*0.,1.,8*0.,2*1.,52*0.,
   ```

9*168.,0.,3*168.,9*0.,.99999,2*0./
This statement initializes the first 229 variables in COMMON block /INPUT/.

8. DATA SAVA/0.,3*1.,16*0.,10*1.,11*.99999,4*0./
This statement initializes 45 variables in COMMON block /INPUT/ beginning with H(1) and ending with DTI. The complete list of variables that accompany /INPUT/ is given in the main program (LOGAM1).

9. DATA SAVG/22*0.,1.,1./
This statement initializes 24 variables in COMMON block /INPUT/ beginning with maintenance policy GA and ending with PIN(6).

10. DATA JSAV/2,4*1/
This statement initializes the 5 locations of the array NIN in common block /INPUT/.

11. DATA SENSY/266*0./
This statement initializes the sensitivity analysis data array. This data is transmitted to LOGAM1 with COMMON block /SENS/.

12. DATA NRU/0/
This statement initializes the LRU counter used in the sensitivity section of LOGAM1.

13. DATA SAV13/5*0.,.08,1.,48.,48.,.9,0.,1.,1.,0./
This statement initializes the last 14 variables in the COMMON block /INPUT/ list.

14. END
This statement terminates the BLOCK DATA routine.

SUBROUTINE BASIC

BASIC computes shipping costs and quantities tied up in pipelines. In general, BASIC uses the rate of flow of LRUs, modules, and parts through the pipes to determine shipping costs and quantities. The flow rates are multiplied by one way shipping costs to find the rate of cost to ship replacements and by two way costs to find the cost to ship repairables. Quantities tied up in the pipelines are computed by multiplying the pipe and delay times with the flow rates.

Shipping costs and quantities tied up in the pipelines are computed using the same statements in BASIC but are made with two separate calls from LOGAM1 (main program).

A. The first section of code includes the subroutine name and the arguments passed from LOGAM1. The remainder of this section assigns computer memeory locations for the variables used in BASIC.

1. SUBROUTINE BASIC (FIXM,FIXP,FIXU,BHF,BHM,BHP,REQD,
   EALR,DALR,GALR,DEPMOD,GMOD,DMOD,TRCE,
   ASLE,ASLO,ASLI)
   These statements include the list of parameters that are input or computed in LOGAM1 and will be used to either determine shipping cost rates or quantities tied up in pipe flow.

2. COMMON /INPUT/ (list of parameters not included)
   These COMMON statements includes a list of input variables of which some are used in BASIC. The list of variables are the same as in LOGAM1.

3. COMMON/BAS/ (list of parameters not included)
   This COMMON statement includes a list of parameters that were computed in LOGAM1 and used in BASIC plus parameters that are computed in BASIC for output to LOGAM1.

4. EQUIVALENCE (statements)
   The EQUIVALENCE statements are not listed here since they are the same as those described in line item A-16 of LOGAM1.

B. This section of BASIC computes either the shipping costs or the demands for repaired modules. The value computed will be for two way (round trip) flow between maintenance levels. The first call from LOGAM1 will compute the shipping cost and the second call will compute the quantity of modules tied up in the repair pipeline.

1. QFME=TMFEO*ASLE

This statement computes a value for repairable modules that are evacuated and returned in the Equipment/Direct Support pipeline. TMFEO is the repairable evacuation flow rate from the organizational level to higher maintenance facilities. TMFEO is the module flow rate (per hour), ASLE is the round trip cost per item (SHTEO) for shipping and round trip pipe length in hours (TEOT) for module demand.

2. QFMO=TMFO*DMOD+TMFOI*ASLO
This statement computes a value for the repairable modules that are evacuated and replaced in the Direct Support/General Support pipeline and also encludes the repair flow (TMFO) at Direct Support when computing module demands. TMFOI is the repairable evacutation flow rate from Direct Support to higher maintenance facilities. ASLO is the round trip cost per item (SHTOI) for shipping and the round trip pipe length in hours (TOIT) for module demand. DMOD is zero for shipping cost. For module demands DMOD is the supply allowance in hours (TUMO) at Direct Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service.

3. QFMI=TMFI*GMOD+TMFID*ASLI
This statement computes a value for the repairable modules that are evacuated and replaced in the General Support/Depot pipeline and also includes the repair flow (TMFI) at General Support when computing module demands. TMFID is the repairable evacutation flow rate from General Support to the Depot. ASLI is the round trip cost per item (SHTID) for shipping and round trip pipe length in hours (TIDT) for module demand. GMOD is zero for shipping cost. For module demand GMOD is the supply allowance in hours (TUMI) for modules at General Support to cover the time between removal of a module from a LRU until the time the module is repaired and returned to service.

4. QFMD=TMFD*DEPMOD
This statement computes only the module demand at the Depot. DEPMOD is zero for the shipping rate costs. TMFD is the rate per hour for modules through the Depot Repair facility. For module demands, DEPMOD is the supply allowance in hours (TUMD) for modules at the Depot to cover the time between removal of a module from an LRU until the module is repaired and returned to service.

C. This section of BASIC computes either shipping costs or mean quantity of demand for scrapped modules. The values computed here are for a one way flow between maintenance facilities. The first call to BASIC computes shipping cost and the second call will compute the module demand.

1. QME=TSME*EALR
This statement computes a value for the flow of modules from Direct Support to Equipment to replace modules scrapped at Equipment. TSME is the modules scrapped per hour per installation. EALR is the cost per item (CDOE) to ship modules one way from Direct Support to Equipment when computing shipping costs. For module demand, EALR is the number of supply hours (REOT) for condemned modules at the Equipment level.

2. QMO=(TSME+TSMO)*DALR
This statement computes a value for the flow of modules from General Support to Direct Support to replace modules scrapped at Direct Support and Equipment. TSME and TSMO are the modules scrapped per hour at Equipment and Direct Support, respectively. For shipping costs, DALR is the cost per item (CDIO) to ship modules from General Support to Direct Support. When computing module demand, DALR is the number of supply hours (ROIT) for condemned modules at Direct Support.

3. QMI=(TSME+TSMO+TSMI)*GALR
This statement computes a value for the flow of modules from Depot to General Support to replace modules scrapped at Equipment, Direct Support, and General Support. TSME, TSMO and TSMI are modules scrapped per hour at the Field facilities. When computing shipping costs, GALR is the cost per item (CDDI) to ship modules from Depot to General Support. For Module demands, GALR is the number of supply hours (RIDT) for condemned modules at the Depot.

4. QMD=(TSME+TSMO+TSMI+TSMD)*FIXM
This statement computes a value for the flow of modules from the factory to the Depot to replace modules scrapped at all maintenance levels. When computing shipping costs, FIXM is the shipping cost (CDFD) per item for modules from factory to Depot. For module demands, FIXM is the reprocurement time in hours (168.*FTM) for modules at the factory.

5. AHPM=BHM-FIXM
This statement computes the discretionary procurement holding time (24.*HPM) in hours for a module. For shipping costs, AHPM is zero.

6. QMDH=(TSME+TSMO+TSMI+TSMD)*AHPM
This statement computes the module demand as a function of discretionary procurement holding time. This term will be zero when computing shipping costs.

D. This section of BASIC computes either the parts demand or the parts shipping cost, as a result of scrap at the Field and Depot

maintenance facilities.

1. QPO=TSPO*DALR
   This statement computes a value for the flow of parts from General Support to Direct Support as a result of parts scrapped at Direct Support. When computing shipping costs, DALR is the cost (CDIO) per item. For parts demand, DALR is the number of supply hours (ROIT) for condemned modules and parts at Direct Support.

2. QPI=(TSPO+TSPI)*GALR
   This statement computes a value for the flow of parts from Depot to General Support as a result of parts scrapped at Direct Support and General Support. When computing shipping costs, GALR is the cost (CDDI) per item. For parts demand, GALR is the number of supply hours (RIDT) for condemned modules and parts at General Support.

3. QPD=(TSPO+TSPI+TSPD)*FIXP
   This statement computes a value for the flow of parts from the factory to the Depot to replace parts scrapped at all maintenance levels. When computing shipping costs, FIXP is the shipping cost (CDFD) per item. For parts demand, FIXP is the reprocurement time (168.*FTP) in hours for parts at the factory.

4. AHPP=BHP-FIXP
   This statement computes the discretionary procurement holding time (24.*HPP) in hours for parts. For shipping costs, AHPP is zero.

5. QPDH=(TSPO+TSPI+TSPD)*AHPP
   This statement computes the parts demand as a function of discretionary procurement holding time for parts scrapped at all maintenance levels.

E. This section of BASIC computes either the shipping cost or the stock demand for replacement of scrapped LRUs. The values computed will be for round trip flow between maintenance levels. TSU in the following statements is the total scrap rate per hour per materiel installation.

1. QUE=TSU*ASLE
   This statement computes a value for the scrap and replacement LRUs in the Equipment/Direct Support pipe. ASLE is the round trip cost (SHTEO) per item for shipping and pipe length time (TEOT) in hours for LRU demands.

2. QUO=TSU*ASLO
   This statement computes a value for the scrap and replacement flow of LRUs in the Direct Support/General

Support pipe. ASLO is the round trip cost (SHTOI) per item for shipping and pipe length time (TOIT) in hours for LRU demands.

3. QUI=TSU*ASLI
This statement computes a value for the scrap and replacement flow of LRUs in the General Support/Depot pipe. ASLI is the round trip cost (SHTID) per item for shipping and pipe length time (TIDT) in hours for LRU demands.

4. QUD=TSU(FIXU+REQD)
This statement computes a value for the flow of LRUs from the factory to the Depot as a result of scrapped LRUs. When computing shipping costs, FIXUs the cost (CDFD) per item and REQD=0. For LRU demands, FIXU is the reprocurement time (168.*FTU) in hours for LRUs at the factory and REQD is the delay time (24.*RDD) in hours between the request time and the handling time.

5. QUDH=TSU*(BHF-FIXU)
This statement computes the LRU demand in the Factory/Depot pipe as a function of discretionary procurement holding time (24.*HPU).

F. This section of BASIC computes either shipping costs or the demand for LRUs as a result of LRUs tied up in the repair float pipelines.

1. QFE=(TUFTO+TUFTI+TUFTD+FGO+FGI+FGD)*ASLE
TUFEC*TC+(TE+TER)*TUFTE
This statement computes a value for the evacuation and replacement of repairable LRUs in the Equipment/Direct Support pipeline. When computing LRU demands the flow of LRUs through the checkout, test, and repair (TUFEC, TUFTE) facilities at the Equipment level are included. When computing shippng costs these flows will cause an error since there is no provision in the statement to zero out the effects of this calculation. For shipping costs, ASLE is the round trip cost (SHTEO) per item. For LRU demands, ASLE is the round trip pipelength (TEOT) in hours.

2. QFO=(TUFTI+TUFTD+FGI+FGD)*ASLO
This statement computes a value for LRU repair float in the Direct Support/General Support pipeline. When computing shipping costs, ASLO is the round trip shipping cost (SHTOI) per item. For LRU demands, ASLO is the round trip pipe length (TOIT) in hours.

3. QFI=(TUFTD+FGD)*ASLI
This statement computes a value for LRU repair float in the General Support/Depot pipeline. When computing shipping costs, ASLI is the round trip shipping cost (SHTID) per

item. For LRU demands, ASLI is the round trip pipe length (TIDT) in hours.

G. This section of BASIC computes either shipping costs or stock quantities to support the replacement of LRUs at a materiel installation. The parameters evaluated are a function of removals and the shipping pipe times between maintenance levels. The removals are represented by the TU term; where TU=(A+F+FNG). The value at each maintenance level is summarized by adding the pipe line value between two levels to the value previously evaluated at the lower level.

1. QYE=TU*TRCE
   This statement will compute stock quantities at the Equipment level to support the removal and replacement of LRUs. Since there are no shipping costs associated with moving LRUs within the Equipment level, QYE will be zero when called from LOGAM1 to compute costs. TRCE is the down time in hours for a materiel system.

2. QYO=TU*(TRCE+ASLE)
   This statement adds the pipeline values (TU*ASLE) for the Equipment/Direct Support pipe to the Equipment level values (TU*TRCE). When computing costs, only the round trip shipping cost between Equipment and Direct Support is included since the shipping cost for QYE=0. For stock quantities summarized at Direct Support, the quantity tied up in the pipe (TU*ASLE) is added to QYE.

3. QYI=QYO+QUO+QFO
   This statement summarizes either stock quantities or shipping costs for LRUs at General Support. The value computed at Direct Support (QYO) is added to the values previously computed for LRU Scrap (QUO) and LRU "float" (QFO) in the General Support/Direct Support pipe.

4. QYD=QYI+QUI+QFI
   This statement summarizes either stock quantities or shipping costs for LRUs at a Depot facility. The value computed at General Support (QYI) is added to the values previously computed for LRU scrap (QUI) and LRU "float" (QFI) in the Depot/General Support pipe.

5. QYF=QYD+QUD
   This statement summarizes either cost or stock quantities for LRUs through the Factory/Depot pipe. The value computed at the Depot level (QYD) is added to the LRU scrap value (QUD) previously computed for the factory/Depot pipe.

6. RETURN
   END
   These statements terminate execution of BASIC and returns

program logic to the main program (LOGAM1).

SUBROUTINE SUPI

This subroutine is called from LOGAM1 to print a supplementary page of output when any of the Equipment level maintenance policies (GI,GC,GJ, or GK) are used. The first reference to SUPI is after the individual LRU output page is printed from LOGAM1. A second reference is made to SUPI after printing a case total page from LOGAM1.

A. This section of SUPI includes the subroutine name, argument list, and the computer memory locations assigned by labled COMMON.

1. SUBROUTINE SUPI(ISUP)
   The argument ISUP in the statement designates the area of code in LOGAM1 from which subroutine SUPI was referenced. ISUP=1 designates that individual LRU data cases are output. ISUP=2 designates that LRU case totals are output.

2. COMMON/INPUT/PUTI(325)
   COMMON/SUPIN/C(117)
   These statements are used to transmit inputs and internal generated data from LOGAM1 to SUPI. C(117) is the data array from which selected parameters will be printed in SUPI.

3. DATA K004FX/6/
   This statement assigns printer output file unit 6 to the WRITE statement parameter K004FX.

4. NAMELIST/BUGSUP/C
   This statement assigns the "C" array from COMMON block /SUPIN/ to NAMELIST /BUGSUP/. BUGSUP is used in this subroutine to print the values of "C" when debugging of a data case is required.

B. This section of SUPI includes the data case debugging logic and selects the level of supplementary output.

1. IBG=PUTI(323)
   This statement sets the debugging flag to the 323rd location of the variable list from COMMON/INPUT/. IBG is a data input variable from NAMELIST/L/.

2. IF(IBG.EQ.1)WRITE(6,BUGSUP)
   This statement prints the values from the variable list of COMMON/SUPIN/ when the debugging flag is turned on in NAMELIST/L/. IBG=1 means that the flag is turned on.

3. IF(ISUP.EQ.2)GO TO 20

This statement transfers logic to statement 20 when summarized outputs are printed. ISUP is equal to 2 in LOGAM1 when case totals are printed. Otherwise ISUP=1 when individual case data is printed in the following statements.

C. This section of SUPI prints data generated at the Equipment level when individual LRU data output is requested.

1. CALL PAGE
   This statement calls subroutine PAGE to print a page header before printing data.

2. WRITE (K004FX,1)
   This statement prints a message describing the outputs to be printed.

3. WRITE (K004FX,2) (C(I),I=106,111)
   This statement prints the manhour and manpower values computed in LOGAM1. The values printed are AEY, BSAEY, AERY, BSAERY, BCAEY, and BCAERY, respectively. These variables are described in part M of the LOGAM1 writeup.

4. IF(PUTI(297).EQ.1)WRITE(K004FX,3) (C(I),I=112,115)
   This statement prints the values for Type I test equipment service demands, test and repair manpower, and manpower self-support. This statement prints output only if the Type I test equipment flag (ETI=1) is on. ETI occupies location 297 of COMMON block /INPUT/ which is also the same location as PIN(5) of the variable list of /INPUT/ in the main program LOGAM1.

5. WRITE(K004FX,4) C(116)
   WRITE(K004FX,5) C(117)
   These statements print the values for the initial provision quantities and cost, respectively, for modules at the Equipment level. The variable names from LOGAM1 for C(116) and C(117) are PQTME and CQTME, respectively.

6. RETURN
   This statement returns to LOGAM1 after an individual LRU data case is printed. This return location will be after the first call to SUPI.

D. This section of SUPI prints summarized data computed at the Equipment level. This code is entered after a case total output is requested in LOGAM1. The output formats are the same formats used in the individual case printouts above.

1. WRITE (K004FX,1)
   This statement prints a message describing the outputs to be printed.

2. WRITE(K004FX,2) (C(I),I=90,95)
   This statement prints the accumulated values for a set (NDLRU) of LRU cases that make up a case total. The values printed here are the accumulated values of the variables printed in line item C-3 above.

3. IF(C(103).EQ.1) WRITE (K004FX,3) (C(I),I=96,99)
   This statement will print the accumulated values for the variables discussed in line item C-4 above. The Type I test equipment flag (STI) is stored in location C(103). When for any individual LRU case is evaluated with Type I test equipment (ETI=1), STI is turned on also.

4. WRITE (K004FX,4) C(100)
   WRITE (K004FX,5) C(101)
   These statements print the accumulated values for the same parameters described in line item C-5 above.

5. RETURN
   END
   These statements return logic flow to the location in LOGAM1 where the second call to SUPI was made.

SUBROUTINE SENSIT

This subroutine is referenced by the main program LOGAM1, when the output pages that are being printed are for a sensitivity analysis case. SENSIT prints a page header that describes the parameters modified for a sensitivity analysis.

A. This section of SENSIT assigns computer memory locations to variables used internal to this routine and to variables that are transmitted by way of labled COMMON from the main program, LOGAM1.

1. DIMENSION VALUE(12)
   This statement sets the maximum size of the array that stores the modified inputs for a sensitivity analysis. The greatest number (MODE) of variables that can be changed per sensitivity case is 12.

2.
```
COMMON/SENS/SENSY(266),NRULE(12),NVAR(12),MODE,
            KPASS,NRU,LRU,NPASS
COMMON/INPUT/VAR(325)
```
   These statements transmit data from the main program, LOGAM1, to SENSIT. These COMMON blocks are discussed in part A of the LOGAM1 description.

3. DATA K001FX/6/
   This statement assigns the printer output file unit 6 to the write statement parameter K001FX.

B. The next section of code stores the value of the parameters modified for a sensitivity analysis and prints a page header for the next page of outputs.

1.
```
   I=0
10 I=I+1
   M=NVAR(I)
   VALUE(I)=VAR(M)
   IF(I.LT.MODE)GO TO 10
```
   These statements loop through the number of input variables (MODE) that were modified, finds the variables location (M) in the input list (VAR), and stores the value (VALUE) for printing.

2. WRITE(K001FX,1)KPASS,NRU,(NVAR(I),VALUE(I),I=1,MODE)
   This statement prints the information for the current LRU data case for which a sensitivity analysis is being performed. The sensitivity logic is discussed in part H of LOGAM1.

3. RETURN
   END
   These statements return logic back to LOGAM1.

SUBROUTINE PAGE

This subroutine prints a header for a page of output. The header describes the logistics analysis performed, the date, and a page number. The information printed here is input in LOGAM1 and is described in line item D-10 of that writeup.

1. IPAGE=IPAGE-1
   This statement decrements the page counter. The LOGAM1 program uses a negative page counter to achieve a particular output effect. As an example the page counter of sample output 9.1.1 is "-29-". The "-29" portion is the negative page counter.

2. WRITE (K001FX,1)IPAGE,TEXT,UNITIS,ANLYIS,REMARK,DATE
   This statement will print header information at the top of an output page.

3. RETURN
   END
   These statements terminate subroutine PAGE and returns control to the routine that called PAGE.

SUBROUTINE IOL

This subroutine computes initial provision quantities. The stock computations are based on the mean demand quantity plus the safety stock coefficient times the square root of the mean demand quantity. The quantities are rounded to whole numbers according to the round off value input for the "Z" parameters. IOL is called from the main program, LOGAM1, when using LOGAM supply rules. A reference is made to IOL for each stock type at each stock provisioning level. Line items J-4, J-5, and J-6 of the LOGAM1 writeup describe the references to IOL.

1. SUBROUTINE IOL(XD,CKK,BQU,BQF,BQT,Z)
   This statement assigns the argument names to match the variable lists from the calling program. The argument definitions are:

   XD - The number of supply points at a provisioning level.
   CKK - The safety stock coefficient.
   BQU - Mean demand quantity of scrapped stock.
   BQF - Mean demand quantity of "float" stock.
   BQT - The computed value for initial provisioning.
   Z - The whole number round off parameter.

2. QUF=XD*(BQU+BQF)
   This statement computes the mean demand quantity for stock at all supply points of a provisioning level.

3. BQT=0.
   IF(XD.LE..5)RETURN
   IF(QUF.LE.0.)RETURN
   These statements return an initial provision quantity of zero if there is not a supply point (XD) at the level being considered or if the mean demand is less than or equal zero.

4. BSQ=SQRT(QUF)
   BQS=CKK*BSQ
   QUFS=BQU+BQF+BQS
   These statements add the safety stock to the mean quantity demand. The safety stock (BQS) is computed as the square root of the mean demand times the safety stock factor.

5. IF(QUFS.LT.0.)QUFS=0.
   BQT=XD*AINT((QUFS/XD)+Z)
   These statements compute the initial provisions for all supply points at a provisioning level. The provisions are computed as whole numbers for each supply point and then multiplied by the number of supply points. The input

parameter "Z" is used as the round off function.

6. RETURN
   END
   These statements return logic back to the calling program. In this case it is LOGAM1.

FUNCTION DEF

This function is referenced from LOGAM1 to compute a Back Order Quantity (BOQ) for unfilled orders of items at the warehouse. The unfilled orders result from an inadequate supply for meeting the demand experienced during any given interval between resupplies. This function is based on a Poisson distribution of demand per resupply interval. The function is called with the expected value stock demand and the stock on hand; whereby, DEF will compute the average number of unfilled orders.

A. The first statement is the function name and it's arguments.

1. FUNCTION DEF(QUF,QT,XD)
   The BOQ value computed by this function is returned to the referencing statement as DEF. The arguments are described as:

   QUF - The expected value demand (quantity) per resupply interval.
   QT - The quantity of stock on hand.
   XD - A multiplication and division factor. In this program XD is always used as 1.0.

B. The next section of code initializes values and checks the input quantities to determine if its necessary to proceed through the Poisson distribution.

1. DEF=0.
   IF(QUF.LE.0.)RETURN
   These statements will return a value of zero for the BOQs if there is not a demand for stock.

2. N=(QT/XD)+.1
   This statement prevents a round off error when converting the quantity of stock on hand to on integer value. XD is equal to 1.0.

3. DEF=QUF
   IF(N.EQ.0)RETURN
   These statements will return a BOQ value equal to the stock demands when there is no stock on hand.

4. X=QUF/XD
   RK=1.
   These statements initialize the expected stock demand and a counter for the number of times the expected stock demand will be halved. If the stock demand is greater than 26,

both the stock demand and the stock on hand are halved until the value for stock demand is not greater than 26. Each time the stock is halved, RK will be doubled. This logic occurs later in this routine. The reasoning is discussed in the next section.

C. The next section of code presets values that will be used in the distribution. A computation is made here to determine if the stocked values are sufficient; if so, a return is made with the value zero for BOQs.

1.
```
5 IF(X.GT.26.)GO TO 3
```
This statement will not permit entering the distribution code until the adjusted stock demand (X) is 26 or less. The logic at statement 3 (line item E-1) will halve both the stock demand and stock on hand until X is 26 or less. The reasoning for this is that the distribution T=EXP(-X) would be insignificant for greater values of X.

2.
```
DEF=0.
IF(N.GT.(2+INT(X+(6.*SQRT(X))))RETURN
```
These statements will return a BOQ value of zero if the adjusted stock quantity is greater than the adjusted distributed value for stock demand.

3.
```
M=N-1
T=EXP(-X)
RN=N
Y=X-RN+(T*RN)
I=0
```
These statements initialize the values that will be used in the next section of code where the BOQ will be computed using the Poisson distribution.

D. The next section of code uses a Poisson distribution to complete the BOQs.

1.
```
6 I=I+1
  RI=I
  T=T*(X/RI)
  Y=Y+(T*(RN-RI))
  IF(I.LT.M)GO TO 6
```
These statements will compute the BOQ (Y) for a segment of the stock on hand quantities (N). The value for Y on entering this logic was for the Nth quantity. This logic will loop until the contribution for each decremented value of N is computed and summed with the previous value of Y.

2. DEF=Y*XD*RK
   This statement multiplies the BOQ value (Y) computed above with the number (RK) of times the stock on hand quantity (QT) was subdivided before entering the Poisson distribution logic. XD is input equal to 1.0. DEF is now the BOQ for the stock demands (QUF) input.

3. IF(DEF.LT.0) DEF=0.
   RETURN
   These statements return to the calling program and DEF will be reset to zero if the computed DEF is a negative value.

E. The following code is used to subdivide the input quantities. A transfer is made here from the statement described in line item C-1 above. The quantities will continue to be segmented (subdivided) until the segmented value of stock demand (X) is less than or equal to 26.

1.
```
3 X=X/2.
  RK=RK+RK
  N=N/2
```
   These statements subdivide the expected stock demand (X) and the stock on hand (N). The number of times the values are segmented is accumulated in RK.

2. DEF=QUF
   IF(N.EQ.0)RETURN
   When the stock on hand segment becomes zero before the expected demand becomes acceptable (26), the BOQ is set to the expected demand that was input (QUF) and a return is made to the calling program.

3. GO TO 5
   This statement transfers logic to statement 5 where another check is made to determine if the stock on hand segment (N) is now less or equal to 26.

4. END
   This statement terminates execution of Function DEF.

SUBROUTINE OPER

This subroutine serves as a post processor once all LRU cases have been processed. OPER is referenced from LOGAM1 when the inputs for IOPER and NU are 1 and -4, respectively. OPER reads Table of Organization and Equipment (TOE) inputs and calls several routines that compute operation and support costs. The routines called are dependent on the TOE data input.

A. This section of OPER contains the assignment of variables in computer memory with labled COMMON blocks and also assigns the TOE data input and output list using NAMELIST.

1. COMMON /T/T(10,200)
   This statement assigns the memory locations for the TOE data that is input to this subroutine.

2. COMMON/POUT/P(16)
   COMMON/ZERO/Z(45)
   COMMON/OFF/O(6)
   COMMON/ENLM/E(7)
   These statements aren't really required in this subroutine. They are actually used in the subroutines called by OPER. The first reference to these COMMON blocks was in LOGAM1 All of the blocks except /ZERO/ are initialized in BLKDAT. COMMON ZERO contains the case totals for cost elements accumulated in LOGAM1. POUT is used in the subroutines referenced by OPER to store O&S cost for output in subroutine EIGHT. COMMON OFF and ENLM are used to store data and personnel numbers for officers and enlisted men, respectively.

3. NAMELIST/TOE/T
   NAMELIST/OPERER/T
   These statements are used to read TOE input data and to print the data, respectively. The description for TOE inputs are found in section G-2 of this document.

B. This section of OPER reads in the TOE data and interrogates the inputs to determine the type of cost to be evaluated. There are eight subroutines called here. Seven of these compute costs based on the type of inputs from NAMELIST/TOE/. The eighth routine will print the O&S cost results. The data are input in fields of 10 values per cost element with a maximum of 200 cost elements. In this section the first location of each set of 10 values is checked to determine which of the subroutines to call to evaluate the cost of the set.

1. LIX=0
   This statement is used in Subroutine SIX to initialize variables on the first pass through the routine.

2. READ(5,TOE)
   This statement reads the input data specified by NAMELIST/TOE/.

3.
```
       J=0
    10 J=J+1
       IF(J.EQ.201) GO TO 20
```
   These statements set and increment the counter for the next cost element to be evaluated. If more than 200 cost elements are entered, a transfer is made to statement 20 where the TOE inputs are printed.

4. IF(T(1,J).EQ.1.)CALL ONE(J)
   This statement will call subroutine ONE to compute total people/category and pay and allowances.

5. IF(T(1,J).EQ.2.)CALL TWO(J)
   This statement will call subroutine TWO to compute personnel related costs.

6. IF(T(1,J).EQ.3.)CALL THREE(J)
   This statement will call subroutine THREE to compute fuel costs.

7. IF(T(1,J).EQ.4.)CALL FOUR(J)
   This statement will call subroutine FOUR to compute ammunition costs.

8. IF(T(1,J).EQ.5.)CALL FIVE(J)
   This statement will call subroutine FIVE to compute instrumentation costs.

9. IF(T(1,J).EQ.6)CALL SIX(J,LIX)
   This statement will call subroutine SIX to compute ARTY/ORD costs, Follow-On-Training (FOT) firing cost for missiles, and adds on cost of instrumentation per firing from subroutine FIVE.

10. IF(T(1,J).EQ.7.)CALL SEVEN(J)
    This statement will call subroutine SEVEN to prepare the output list in the proper units of cost.

11. IF(T(1,J).EQ.8.)CALL EIGHT(J)
    This statement will call subroutine EIGHT to print the operating and support costs.

12. IF(T(1,J).EQ.8.)GO TO 30

A value of 8 in the first location of a cost element field of TOE terminates the post processor evaluation. After printing the outputs, logic is transferred to statement 30 where a return is made to LOGAM1.

13. GO TO 10
This statement "loops" back to statement 10 where the cost element counter is incremented and the evaluation for the next set of TOE data will begin.

14. 20 WRITE(6,OPERER)
This statement prints the TOE data when more than 200 cost elements (fields of 10 values) are input. NAMELIST/TOE/ could just as well have been used instead of OPERER.

15. STOP
This statement will stop program execution once an input error is detected.

16. 30 RETURN
END
These statements signal a normal termination to the TOE evaluation and will return logic back to LOGAM1.

SUBROUTINE ONE

This subroutine is called from subroutine OPER to compute total people/category and pay and allowances.

A. This section of ONE contains the assignment of variables in computer memory with labeled COMMON blocks and also assigns TOE and other data to NAMELISTs for printing.

1.
```
SUBROUTINE ONE(J)
```
This statement is the entry point for the subroutine. ONE is called from subroutine OPER when the value of J=1. J is the first in a ten element field of TOE data.

2.
```
COMMON/POUT/P(16)
COMMON/T/T(10,200)
COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
COMMON/ZERO/Z(75)
COMMON/ENLM/E(7)
COMMON/OFF/O(6)
```
These common statements have been described previously in other subroutines. INPUT is described in part A of the LOGAM1 description and the others are described in part A of subroutine OPER.

3.
```
REAL MAINT, NMAINT, NCREW, NDED, DED
```
This statement assigns floating point status (real) to these variables that would ordinarily be integers.

4.
```
NAMELIST/ONER/T
NAMELIST/BUGONE/J,P,E,O
```
These statements assign TOE inputs and TOE evaluated data to NAMELIST names for output when debugging of the data is required.

B. This section of ONE interrogates a "personnel count" entry of TOE data and sets the assignments for crew, maintenance, and dedicated individuals. The logic to transfer to either officer or enlisted men evaluations is included here.

1.
```
MAINT=AB(T(9,J))
NMAINT=1.-MAINT
```
These statements set the maintenance personnel multipliers. If T(9,J)=1., then the personnel assigned to TOE level J is for maintenance. When T(9,J)=0., the assignment will be for non-maintenance (NMAINT=1) personnel.

2.
```
CREW = AB(T(7,J))
NCREW=1. - CREW
```
These statements set the crew personnel multipliers. If

T(7,J)=1., then the personnel assigned to TOE level J is for crew. When T (7,J)=0., the assignment will be non-crew (NCREW=1.) personnel.

3. DED=AB(T(5,J)+T(6,J))
NDED=1.-DED
These statement set the dedicated and/or overhead personnel multipliers. If either the dedicated flag T(6,J) or the overhead flag T(5,J) are on (=1), the dedicated multiplier DED will equal 1. NDED is computed here but it is never used in the program.

4. IF(T(3,J).EQ.0.) GO TO 10
IF(T(3,J).EQ.1.) GO TO 20
WRITE (6, ONER)
STOP
These statements check the officer/enlisted men flags to determine which section of this program to transfer to for personnel cost computations. T(3,J)=0 will transfer logic to statement 10 to evaluate officer data. T(3,J)=1 will transfer logic to statement 20 to evaluate the enlisted men costs. Any other value for T(3,J) will result in the printing of the TOE inputs and termination of program execution.

C. The next section of ONE computes the pay and allowance for officers/WOs. The computations include costs for CREW, maintenance, and support that are accumulated for all sets of TOE data. The user should be aware that to input both the crew and maintenance flags equal to 1 will result in a double accounting of dedicated/overhead costs. This will also be true for the costing of enlisted men. The statements for O(4) and O(5) for officers and E(4) for enlisted men are the ones impacted. In the following statements T(2,J) is the quantity of officers and T(10,J) is the cost per officer per year.

1. 10 O(01)=O(01)+CREW*T(2,J)
This statement accumulates the number of officers assigned to crew.

2. O(02)=O(02)+NCREW*T(2,J)
This statement accumulates the number of officers that are not assigned to the crew. This will be the number of officers charging indirect.

3. O(03)=O(03)+DED*NCREW*NMAINT*T(10,J)*T(2,J)
This statement accumulates the total annual pay and allowance for dedicated organizational/overhead officers less crew and maintenance officers.

4. O(04)=O(04)+DED*MAINT*T(10,J)*T(2,J)
This statement accumulates the total annual pay and

allowance for all dedicated/overhead maintenance officers.

5. O(5)=O(05)+DED*CREW*T(10,J)*T(2,J)
This statement accumulates the total annual pay and allowance for all dedicated/overhead officers assigned to the crew.

6. O(6)=O(06)+T(2,J)
This statement accumulates the total number of officers in the TOE organization.

7. P(01)=P(01)+T(10,J)*CREW*PUTI(225)*T(2,J)
This statement accumulates the total O&S lifecycle pay and allowance for officers assigned to crew into the total direct P&A array. PUTI (225) is equivalent to YR in COMMON block INPUT, the number of O&S years.

8. P(02)=P(02)+T(10,J)*NCREW*PUTI(225)*T(2,5)
This statement accumulates the total O&S lifecycle pay and allowance for officers not assigned to crew into the total non-direct P&A array. PUTI(225) is equivalent to YR.

9. IF (IBUG.EQ.1) WRITE (6,BUGONE)
This statement will print the values of the variable list in NAMELIST/BUGONE/when the debugging flag IBG is turned on (=1) in the NAMELIST/L/input data.

10. RETURN
This statement returns program logic control back to subroutine OPER.

D. The following section of code computes the pay and allowance for enlisted men. The variables used here are the same as those used above for officers except that the results are accumulated into "E" instead of "O".

1. 20 E(01)=E(01)+CREW*T(2,J)
This statement accumulates the number of enlisted men that are assigned to crew.

2. E(02)=E(02)+NCREW*NMAINT*T(2,J)
This statement accumulates the number of enlisted men at the organization that are neither crew nor maintenance personnel.

3. E(03)=E(03)+DED*NCREW*NMAINT*T(10,J)*T(2,J)
This statement accumulates total annual pay and allowance for all overhead/dedicated organizational enlisted men.

4. E(04)=E(04)+DED*MAINT*T(10,J)*T(2,J)
This statement accumulates the total annual pay and allowance for all overhead/dedicated maintenance enlised

men.

5. E(05)=E(05)+CREW*T(10,J)*T(2,J)
   This statement accumulates the total annual pay and allowance for all crew enlisted men.

6. E(06)=E(06)+T(2,J)*DED*MAINT
   This statement accumulates the total number of overhead/dedicated enlisted men in organizational maintenance.

7. E(07)=E(07)+T(2,J)*NMAINT
   This statement accumulates the total number of enlisted personnel that are non-maintenance (dedicated, overhead and crew).

8. P(01)=P(01)+T(10,J)*CREW*PUTI(225)*T(2,J)
   This statement accumulates the total O&S lifecycle pay and allowance for enlisted crew into the total direct P&A array. PUTI(225) is equivalent to YR.

9. P(02)=P(02)+T(10,J)*NCREW*MAINT*PUTI(225)*T(2,J)
   This statement accumulates the total O&S lifecycle pay and allowance for organizational maintenance enlisted personnel into the non direct P&A array. PUTI (225) is equivalent to YR.

10. IF(IBUG.EQ.1) WRITE (6,BUGONE)
    This statement prints the values for the parameter list of NAMELIST/BUGONE/ when debugging of a data case is requested (IBG=1).

11. RETURN
    END
    These statements terminate subroutine ONE execution and returns control to subroutine OPER.

SUBROUTINE TWO

This subroutine is a part of the past processing section of LOGAM1 that computes operating and support costs. This subroutine computes personnel related costs for both officers and enlisted men.

A. The first section of TWO includes the subroutine name and the assignment of variables in computer memory.

1. SUBROUTINE TWO(J)
   This statement is the entry point for the subroutine. TWO is called from subroutine OPER when the value of J=2. J is the first element in a 10 element field of TOE data.

2. COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
   COMMON/T/T(10,200)
   COMMON/ZERO/Z(75)
   COMMON/ENLM/E(7)
   COMMON/OFF/O(6)
   COMMON/POUT/P(16)
   These statements assign variable names to computer memory. They are used to transfer data between subroutines. The description for these statements can be found in part A of LOGAM1 and part A of OPER.

3. NAMELIST/BUGTOO/J,P,E,O
   NAMELIST/TWOER/T,J,P,E,O
   These statements set up a parameter list of TOE and personnel related data to output when debugging of a data case is required or when an error is encountered in the TOE inputs.

B. The next section of TWO sums the LOGAM maintenance people and checks the value of the second element in a TOE field, T(2,J), to determine the area of TWO to transfer to. There are three possible values for T(2,J). T(2,J)=1 signifies that the associated input field is for enlisted men. T(2,J)=2 and its associated data will compute costs for replacements, transients, and quarters for officers. T(2,J)=3 will cost officer contribution to permanent change of station.

1. QEPM=Z(66)+Z(68)+Z(69)+Z(70)+Z(71)+Z(72)+Z(73)
   This statement computes the number of field level enlisted test and repair personnel as determined by the LOGAM1 logistics support calculations. The "Z" values were computed in section U-5 through U-13 of LOGAM1 as PERS. The assignments for the "Z" values are:

   Z(66)=PERS(1,1) - Fault isolate and test at E
   Z(67)=PERS(2,1) - Test and checkout at E

```
Z(68)=PERS(3,1) - Test and checkout at DS
Z(69)=PERS(4,1) - Test and checkout at GS
Z(70)=PERS(5.1) - Test and checkout at Depot
Z(71)=PERS(1,2) - Remove and replace at E
Z(72)=PERS(2,2) - Repair at E
Z(73)=PERS(3,2) - Repair at DS
Z(74)=PERS(4,2) - Repair at GS
Z(75)=PERS(5,2) - Repair at Depot
```

Since QEPM is for personnel at the field level, Z(70) should not be included; whereas Z(67) and Z(74) should be added.

2. ```
   If(T(2,J).EQ.1) GO TO 10
   IF(T(2,J).EQ.2) GO TO 20
   IF(T(2,J).EQ.3) GO TO 30
   ```
   These statements will transfer logic to either statement 10, 20,, or 30 to evaluate TOE inputs for enlisted men, officer replacements, and officer PCS, respectively.

3. ```
   WRITE(6,TWOER)
   STOP
   ```
   This statement prints TOE input and computed values when an unacceptable number is input for T(2,J). Any value other than a 1,2 or 3 for T(2,J) will cause this print and termination of program execution.

C. This section of TWO computes the operation and support costs contributed by enlisted men to PCS, replacement, other direct, and other indirect costs. PUTI (225) in the following statements is equivalent to YR of LOGAM1. YR is the number of years of operation and support.

1. ```
   10 PSCE=(QEPM+E(7))*T(3,J)*T(4,J)*PUTI(225)
      P(3)=P(3)+PSCE
   ```
   These statements compute the O&S cost of permanent change of station for enlisted men and accumulate the value with other PCS costs (P(3)). T(3,J) and T(4,5) are the rate of change of station and the cost per change, respectively. E(7) is the number of non-maintenance enlisted men.

2. ```
   P(6)=P(6)+T(5,J)*PUTI(225)
   ```
   This statement computes other direct O&S cost of the organization for enlisted men. T(5,J) is the cost per year.

3. ```
   CEPRC=E(1)*T(6,J)*T(7,J)*PUTI(225)
   ```
   This statement computes the O&S cost for enlisted crew attrition. E(1) is the number of enlisted crew personnel, T(6,J) is the attrition rate, and T(7,J) is the replacement cost per man including training.

4. ```
   ODEPRC=E(2)*T(8,J)*T(9,J)*PUTI(225)
   ```
   This statement computes the O&S attrition cost for enlisted

personnel at organizations that are neither crew nor maintenance. E(2) is the number of enlisted personnel at organization, T(8,J) is the attrition rate, and T(9,J) is the replacement cost per man.

5. P(7)=P(7)+CEPRC+ODEPRC
This statement adds the attrition cost for enlisted crew and organization to personnel replacement cost accumulation.

6. P(11)=P(11)+T(10,J)*PUTI(225)
This statement accumulates O&S "other indirect" costs. T(10,J) is other direct costs per year.

7. IF(IBUG.EQ.1) WRITE (6,BUGTOO)
This statement prints the O&S evaluations from the post processing routines when debugging of the output is required.

8. RETURN
This statement terminates the enlisted personnel computations and returns logic back to subroutine OPER.

D. This section of TWO computes the O&S cost contributed by officers for replacements, transients, and quarters. PUTI(225) is YR, the number of O&S years. This section of code also includes the cost of enlisted personnel, transients, and quarters.

1. 20 CORC=O(1)*T(3,J)*T(4,J)*PUTI(225)
OORC+O(2)*T(5,J)*T(6,J)*PUTI(225)
These statements compute the O&S attrition cost for officers assigned to crew and to organization, respectively. O(1) and O(2) are the number of officers, T(3,J) and T(5,J) are the attrition rates, and T(4,J) and T (6,J) are the cost per replacement.

2. P(7)=P(7)+CORC+OORC
This statement adds the officer attrition costs to the accumulated O&S replacement cost (P(7)).

3. CTPPD=O(3)*T(7,J)+E(3)*T(8,J)
This statement computes the transient cost for officers and enlisted men. O(3) and E(3) are the annual pay and allowance for dedicated organizational/overhead officers and enlisted men, respectively. T(7,J) and T(8,J) are the percentages of P&A that are charged for transient cost.

4. CTPPM=O(4)*T(7,J)+(E(4)+P(16)/PUTI(225))*T(8,J)
This statement computes transient costs as a factor of pay and allowance for dedicated/overhead maintenance personnel for officers and enlisted men, respectively. T(7,J) and T(8,J) are the factors. P(16) was computed in LOGAM1 as the manpower cost of operating test equipment.

5. CTPPC=O(5)*T(7,J)+E(5)*T(8,J)
This statement computes the transient cost of officers and enlisted men as a factor of the pay and allowance for dedicated/overhead crew. T(7,J) and T(8,J) are the percentage factors.

6. P(8)=P(8)+(CTPPD+CTPPM+CTPPC)*PUTI(225)
This statement adds the transient costs into the accumulated O&S output array.

7. CQMUO=O(6)*T(9,J)*PUTI(225)
CQMUE=E(6)*T(10,J)*PUTI(225)
These statements compute the cost of living quarters for officers and enlisted men, respectively. O(6) and E(6) are total officers and enlisted men in the TOE organization. T(9,J) is the annual cost per officer and T(10,J) is the average annual cost per enlisted men.

8. P(9)=P(9)+CQMUD+CQMUE
This statement sums the cost of quarters for all organizational personnel.

9. IF(IBUG.EQ.1) WRITE(6,BUGTOO)
This statement prints the O&S post processor arrays when a debugging request is made.

10. RETURN
This statement terminates execution of this subroutine and returns logic to OPER.

E. This section of Code computes the contribution of officers to the cost of permanent change of stations. PUTI(225) in the following statements is the number of years of O&S. The cost of medical support is also computed in this section.

1. 30 PCSO=T(3,J)*T(4,J)*PUTI(225)*O(6)
These statements computes the cost of permanent change of station for officers. T(3,J) and T(4,J) are the rate of change and the cost per officer, respectively. O(6) is the number of officers in a TOE organization.

2. P(3)=P(3)+PCSO
This statement adds the permanent change of station cost for officers into the accumulated array for all PCSs.

3. P(10)=P(10)+(O(6)+QEPM+E(7))*T(5,J)*PUTI(225)
This statement computes the cost of medical support. O(6), QEPM, and E(7) are the number of TOE organization officers, number of field level test and repair personnel, and the number of enlisted men that are non-maintenance, respectively. T(5,J) is the average annual cost per person

for medical support.

4. IF(IBUG.EQ.1) WRITE (6,BUGTOO)
   This statement prints the O&S cost post processor arrays when a debugging request is made.

5. RETURN
   END
   These statements terminate execution of the subroutine and transfers logic to OPER.

SUBROUTINE THREE

This subroutine computes the O&S cost of Petroleum, Oil and Lubricant (POL). THREE is called from Subroutine OPER when the value for T(1,J)=3.

A. This section of THREE includes the entry point (subroutine name), the common assignments, and the NAMELIST debugging statement.

1. SUBROUTINE THREE (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. There are a maximum of 200 line items with each line item having 10 possible input values.

2. COMMON/T/T(10,200)
   COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
   COMMON/POUT/P(16)
   These statements are used to transmit data between subroutines. /INPUT/ is described in part A of LOGAM1. /T/ and /POUT/ are described in part A of subroutine OPER.

3. NAMELIST/BUGTRI/T,J,P
   This statement provides a list of TOE inputs and outputs to print when debugging of output is requested.

B. This section of code computes the POL cost and terminates execution of the subroutine.

1. P(4)=P(4)+T(10,J)*T(4,J)*T(5,J)*T(6,J)*
         T(7,J)*PUTI(225)
   This statement computes the cost of petroleum, oil, and lubricants for a specific item of equipment and accumulates the cost into the POL element (P(4)) of the output array. The other parameters used in the equation are:

T(10,J)	Quantity of TOE line item using the fuel.
T(4,J)	Hours per year that the line item is used.
T(5,J)	Rate of fuel usage for the line item.
T(6,J)	Cost of fuel per usage rate unit.
T(7,J)	Fractional increase over fuel use to allow for oil and lube.
PUTI(225)	Number of O&S years (YR from LOGAM1)

2. IF(IBUG.EQ.1) WRITE (6,BUGTRI)
   This statement prints the TOE inputs and the O&S output array from NAMELIST/BUGTRI/ when a debugging request is made

in LOGAM1 inputs.

3. RETURN
   END
   These statements terminate execution of subroutine THREE and returns logic to subroutine OPER.

SUBROUTINE FOUR

This subroutine computes the O&S cost of ammunition. Subroutine FOUR is called from subroutine OPER when the TOE input T(1,J)=4.

A. This section of subroutine FOUR includes the entry point (subroutine name), the COMMON assignments, and the NAMELIST debugging statement.

1. SUBROUTINE FOUR (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. For TOE inputs there is a maximum of 200 line items with each line item having 10 possible input values.

2. COMMON/POUT/P(16)
   COMMON/INPUT/PUTI(322),IBUG, PUTO(2)
   COMMON/T/T(10,200)
   These statements are used to transmit data between subroutines. /INPUT/ is described in part A of LOGAM1. The COMMON blocks /POUT/ and /T/ are the O&S output and TOE input arrays, respectively. These two COMMON blocks are described in part A of subroutine OPER.

3. NAMELIST/BUGFOR/T,J,P
   This statement provides a list of TOE inputs and outputs to print when a debugging request is made from LOGAM1.

B. This section of FOUR computes the cost of ammunition during the O&S phase of a program and terminates subroutine execution.

1. P(5)=P(5)+(T(2,J)+T(3,J)+T(4,J)+T(5,J))*PUTI(225)
   This statement computes the cost of ammunition for a TOE line item and accumulates this cost into the output array (P(5)). The factors that make up ammunition cost are:

T(2,J)	Cost per year to transport ammunition.
T(3,J)	Cost per year associated with ammo firings for unit training.
T(4,J)	Cost per year for range support of ammo firings.
T(5,J)	Cost per year for A.P.L. data takers associated with firings.
PUTI(225)	Number of O&S years. Same as YR in the LOGAM1 routine.

2. IF(IBUG.EQ.1)WRITE(6,BUGFOR)
   This statement prints the TOE inputs and the O&S output array from NAMELIST/BUGFOR/ when a debugging request is made from LOGAM1.

3. RETURN
   END
   These statements terminate the execution of subroutine FOUR and returns control to subroutine OPER.

SUBROUTINE FIVE

This subroutine sums instrumentation costs temporarily into the output array, P(12), for later use in subroutine SIX. FIVE is called from subroutine OPER when the value for T(1,J)=5.

A. This section of FIVE includes the subroutine name, COMMON assignments, and the NAMELIST debugging statement.

1. SUBROUTINE FIVE (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. There are a maximum of 200 line items with each line item having 10 possible input values.

2. COMMON/POUT/P(16)
   COMMON/T/T(10,200)
   COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
   These statements are used to transmit data between subroutines. COMMON /INPUT/ is described in part A of LOGAM1. The COMMON blocks /POUT/ and /T/ are the TOE output and input arrays, respectively. These two COMMON blocks are described in part A of subroutine OPER.

3. NAMELIST/BUGFIV/T,J,P
   This statement provides a list of TOE inputs and outputs to print when a debugging request is made from LOGAM1.

B. This section of FIVE sums the instrumentation costs into location P(12) of the O&S printout array and terminates the subroutine execution.

1. ```
      DO 10 K=2,10
   10 P(12)=P(12)+T(K,J)
   ```
   This DO loop accumulates the 9 possible instrumentation cost inputs for line item J into output array element P(12).

2. IF(IBUG.EQ.1) WRITE (6,BUGFIV)
   This statement prints the TOE inputs and the O&S output array from NAMELIST/BUGFIV/ when a debugging request is made from LOGAM1.

3. RETURN
   END
   These statements terminate execution of this subroutine and transfers control to subroutine OPER.
   ```

SUBROUTINE SIX

This subroutine computes Artillery/Ordnance and Follow-On-Training (FOT) firing costs for missiles. The instrumentation cost summed in subroutine FIVE is used to compute ARTY/ORD costs. SIX is called from subroutine OPER when the value of T(1,J)=6.

A. This section of SIX includes the subroutine name, COMMON assignments, and debugging NAMELIST statements.

1. SUBROUTINE SIX (J,LIX)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. For TOE inputs there is a maximum of 200 line items with each line item having 10 possible input values. LIX is an initialization flag used to reset the temporary location P(12) for instrumentation cost.

2. COMMON/POUT/P(16)
   COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
   COMMON/T/T(10,200)
   These statements are used to transmit data between subroutines. COMMON /INPUT/ is described in part A of LOGAM1. The COMMON blocks /pout/ and /T/ are the O&S output and TOE input arrays, respectively. These two common blocks are described in part A of subroutine OPER.

3. NAMELIST/SIXER/T,J,P
   NAMELIST/BUGSIX/T,J,P
   These statements provide a list of the O&S data to be printed if an error occurs in the TOE inputs or a debugging request is made in LOGAM1.

B. This section of SIX stores the instrumentation costs, resets the temporary location used for instrumentation and determines whether to cost ARTY/ORD or FOT.

1. IF(LIX.EQ.1) GO TO 30
   LIX is set to zero in subroutine OPER so that on the first call to subroutine SIX, the instrumentation cost of missiles will be stored into a new variable and the temporary location P(12) can be reset. This occurs only one time per TOE evaluation because LIX is set to one in this subroutine and the resetting logic will be bypassed on the next call to SIX. For missile instrumentation costs to be included in SIX, the TOE line item data for instrumentation must first be evaluated in subroutine FIVE.

2. SIGIN=P(12)
   This statement stores the missile instrumentation cost that was evaluated in subroutine FIVE into the local variable

SIGIN.

3. P(12)=0. This statement resets the temporary location for the instrumentation cost.

4. LIX=1
This statement sets the instrumentation cost flag so that the two statements above will never be entered again.

5.
```
30 IF(T(2,J).EQ.1.) GO TO 10
   IF(T(2,J).EQ.2.) GO TO 20
```
These statements check the second input value of a TOE line item to determine if the input is for ARTY/ORD or FOT costs. A value of 1 will transfer logic to do ARTY/ORD costs and a value of 2 will cost FOT.

6.
```
WRITE(6,SIXER)
STOP
```
Any value for T(2,J) other than a 1 or 2 is an error condition that will result in the printout of the variables in NAMELIST/SIXER/ and termination of program execution.

C. This section of SIX computes the Artillery/Ordnance costs during the O&S phase of a program. This logic is executed when T(2,J)=1.

1. 10 CATAO=T(3,J)*T(10,J)
This statement computes the annual cost of transport for ARTY/ORD firings. T(3,J) is the cost of transport per firing and T(10,J) is number of firings per year.

2. CAMIAO=T(10,J)*SIGIN
This statement computes the annual missile instrumentation costs where SIGNIN is the instrumentation cost per firing.

3. CARSUO=T(10,J)*(T(5,J)+T(6,J))
This statement computes the annual cost of range support and use. T(5,J) is the support cost per firing and T(6,J) is cost of range use per firing.

4. CACSAO=T(10,J)*T(7,J)
This statement computes the annual contractor support for ARTY/ORD firings. T(7,J) is the cost of support per firing.

5. P(5)=P(5)+(CATAO+CAMIAO+CARSUO+CACSAO+T(4,J))*PUTI(225)
This statement computes the ARTY/ORD firing costs over the O&S phase and accumulates this cost into the unit training AMMO and missile cost P(5). T(4,J) is the cost per year for APL data takers associated with firings and PUTI (225) is the number of O&S years.

6. IF(IBUG.EQ.1) WRITE (6,BUGSIX)
This statement prints the TOE inputs and O&S outputs from

the NAMELIST/BUGSIX/ data list when a debugging request is made in LoGAM1.

7. RETURN
This statement terminates subroutine SIX execution and returns program control to subroutine OPER.

D. This section of SIX computes the Follow-on-Training firing costs. T(10,J) in the following statements is the number of firings per year.

1. 20 CATFOT=T(10,J)*T(3,J)
This statement computes the annual cost of transportation for FOT firings. T(3,J) is the cost per firing.

2. CAIFOT=T(10,J)*SIGIN
This statement computes the annual instrumentation costs for FOT firings. SIGIN is the instrumentation cost per firing.

3. CARSUF=T(10,J)*(T(5,J)+T(6,J))
This statement computes the cost of FOT range support and use. T(5,J) is the cost per firing for range support and T(6,5) is the annual cost per firing for range use.

4. CACSFO=T(10,J)*T(7,J)
This statement computes the annual cost of FOT contractor support where T(7,J) is the cost per firing.

5. CAPFOT=T(10,J)*T(8,J)
This statement computes the annual cost of overseas missile preparation where T(8,J) is the cost per FOT firing.

6. P(5)=P(5)+(CATFOT+CAIFOT+CARSUFT+CACSFO+CAPFOT+T(4,J))*
PUTI(225)
This statement computes the FOT firing costs over the O&S phase and accumulates this cost into the unit training AMMO and missile cost P(5). T(4,5) is the annual cost for data takers associated with firings. PUTI(225) is the number of O&S years.

7. IF(IBUG.EQ.1)WRITE(6,BUGSIX)
This statement prints the TOE inputs and the O&S outputs from the NAMELIST/BUGSIX/ data list when a debugging request is made from LOGAM1.

8. RETURN
END
These statements terminate execution of subroutine SIX and returns program control to subroutine OPER.

SUBROUTINE SEVEN

This subroutine prepares the output list (P) in the proper units of cost. SEVEN is called from subroutine OPER when the value of T(1,J)=7. A value of 7 should be input to TOE just prior to printing the outputs but only if the cost units of TOE need to be converted to the LOGAM units. AMULT from LOGAM1 is used in the conversion.

A. The next section of SEVEN contains the assignment of variables in computer memory using labled COMMON blocks and assigns the output list P to a NAMELIST name for debugging purposes.

1.
```
COMMON/INPUT/PUTI(322),IBUG,PUTO(2)
COMMON/ZERO/Z(75)
COMMON/POUT/P(16)
COMMON/DAPAM/SDA(10),WD(5)
```
These statements have been described previously in other subroutines. INPUT and DAPAM are described in part A of LOGAM1. ZERO and POUT descriptions can be found in part A of subroutine OPER.

2.
```
NAMELIST/BUGSEV/P
```
This statement assigns the TOE ouput data array P to NAMELIST BUGSEV for printing when debugging of a data case is required.

B. The next section of SEVEN converts cost to the proper units, adds outputs from LOGAM to TOE outputs, and totals the costs.

1.
```
   DO 10 I=1,11
10 P(I)=P(I)*P(14)
```
This DO loop uses AMULT, the conversion factor from LOGAM1, to put the TOE costs in the proper units. AMULT and P(14) occupy the same location in COMMON/POUT/.

2.
```
P(6)=P(6)+SDA(10)
```
This statement adds the cost of maintaining a line item in the supply system (SDA(10)) to "other direct" costs. SDA(10) is computed in LOGAM1 as F10, where F10 is the total case accumulation of U17.

3.
```
P(7)=P(7)+SDA(8)
```
This statement adds the cost of training test and repair personnel (SDA(8)) to the "personnel replacement" costs. SDA(8) is computed in LOGAM1 as F8, where F8 is the total case accummulation of U12.

4.
```
P(11)=P(11)+SDA(9)
```
This statement adds the cost of housing test equipment and shipping of mod kits (SDA(9)) with "other indirect" costs. SDA(9) is computed in LOGAM1 as F9, where F9 is the total

case accumulation of CSHTF+CFT.

5.
```
P(12)=P(15)
```
This statement stores the maintenance support grand total cost P(15) from LOGAM1 into another location of the post processor output array. P(15) was computed in LOGAM1 as OPERSV.

6.
```
   P(13)=0
   DO 20 I = 1,12
20 P(13)=P(13)+P(I)
```
These statements accumulate the post processor outputs into a grand total.

7.
```
IF(IBUG.EQ.1) WRITE (6,BUGSEV)
```
This statement will print the post processor output array (P) when debugging of a data case is requested.

8.
```
RETURN
END
```
These statements terminate the execution of subroutine SEVEN and returns logic to subroutine OPER.

SUBROUTINE EIGHT

This subroutine prints the System Maintenance Support and the System Operating and Support Costs. The Maintenance Support costs are printed when Subroutine EIGHT is called from LOGAM1. Each time a LRU case total or grand total is printed in LOGAM1 a call is made to Subroutine EIGHT to print the Maintenance Support costs.

This subroutine is also called by Subroutine OPER, when the TOE input T(1,J)=8, to print the system operating and support costs that were evaluated in the post processor section of this program.

A. This section of EIGHT includes the subroutine name and COMMON assignments.

1. SUBROUTINE EIGHT(J)
   The argument "J" is the TOE line item number when EIGHT is called from subroutine OPER. The value of J will be one greater than the line item number for the last cost evaluation. For example, if the last cost evaluation was for line item T(1,75) then J will be 76 when EIGHT is called. The value of J is set to 3 when called from LOGAM1. It is unlikely, but there is one instance where this could cause a problem. If the number of TOE line items costed is 2 then J will be 3 when EIGHT is called, therefore, causing the Maintenance Support costs to be printed rather than the O&S costs.

2. COMMON/POUT/P(16)
   COMMON/BX/X1,X2,X3,X4,X5,X6,X7
   COMMON/DAPAM/SDA(10),WD(6)
   COMMON/HEADER/DATE(3),TEXT(48),RK(26),IPAGE
   These statements are used to transfer data between subroutines. /POUT/ is described in part A of subroutine OPER while the others are described in part A of LOGAM1.

3. DIMENSION PER(25)
   This statement dimensions an array for storing individual support costs as a percentage of the total support cost.

4. WRITE (6,160)
   This statement will cause the line printer to skip to the top of the next page.

B. This section of EIGHT determines the level of support costs to print and prints the page titles for the level selected. For the maintenance support level, LOGAM1 outputs are stored into the printout array (P).

1. IF(J.EQ.3) GO TO 250
   When J=3 the reference to EIGHT was made from LOGAM1,

therefore, go to statement 250 and set LOGAM1 values into the P array.

2. GO TO 251
The value of J is not a 3, go to statement 251 and print System Operation and Support cost page header.

3.
```
250 P(6)=SDA(10)
    P(7)=SDA(8)
    P(11)=SDA(9)
    P(12)=0.
    P(13)=SDA(8)+SDA(9)+SDA(10)
```
These statements store LOGAM1 maintenance support costs into the output array. The SDA values are the accumulated costs of F8,F9, and F10 which are described in part P-2 of LOGAM1.

4. WRITE (6,161)IPAGE,TEXT
This statement prints the page header and title for the LOGAM System Maintenance Support Cost outputs.

5. 251 CONTINUE
This statement is transfer point when the outputs will be for the Operation and Support costs.

6. IF(J.NE.3) WRITE (6,162) IPAGE,TEXT
This statement will print the page header and title for System Operation and Support Costs. A call from subroutine OPER initiates this print.

C. The next section of code computes cost totals and percentages of totals for R&D, Investment, O&S, and grand total.

1.
```
    TAI=0
    DO 40 I=1,7
40  TAI=TAI+SDA(I)
    X=P(13)-P(12)+TAI
```
These statements compute the total O&S costs. The SDA values are computed in LOGAM1 as F1-F7. P(13) is either the maintenance support costs described in section B-3 above or the total support cost from the post processor subroutine SEVEN. P(12) is set to LOGAM1's OPERSV in subroutine SEVEN.

2. IF (X.EQ.0.) X=1.0
This statement sets the O&S phase cost to 1. in order that a division by zero will not occur when percentages of total cost are computed.

3. PER(1)=100.
This statement sets the percentage of R&D cost to 100 percent since there is only one cost element (Development Engineering) output for this cost phase.

4.
```
   XB=0.
   DO 85 I=2,6
85 XB=XB+WD(I)
```
These statements sum the individual Investment Costs into a total Investment. WD are the individual costs that are accumulated in LOGAM1 as F12-F16.

5.
```
IF (XB.EQ.0)XB=1.
```
This statement sets the total Investment Cost to 1. to prevent dividing by zero while computing individual cost percentages.

6.
```
GT=X+XB+WD(1)
```
This statement sums the O&S costs (X), Investment Costs (XB), and Development Cost (WD(1)) into a grand total.

7.
```
   DO 86 I=2,5
86 PER(I) = (WD(I)/XB)*100.0
   PER(25) = (WD(6)/XB)*100.0
```
These statements compute the percentage of individual Investment Costs to the total Investment Cost.

8.
```
PER(6)=(P(1)/X)*100.0
PER(7)=(SDA(1)/X)*100.0
PER(8)=(P(2)/X)*100.0
PER(9)=(P(3)/X)*100.0
```
These statements compute the percentage of individual O&S military personnel costs to the total O&S cost.

9.
```
PER(10)=(SDA(2)/X)*100.0
PER(11)=(P(4)/X)*100.0
PER(12)=(P(5)/X)*100.0
```
These statements compute the percentage of individual O&S Consumption Costs to the total O&S cost.

10.
```
   DO 87 I=13,17
87 PER(I)=(SDA(I-10)/X)*100.0
   DO 88 I=18,23
88 PER(I)=(P(I-12)/X)*100.0
```
These statements compute the percentage of the individual direct and indirect support operations cost to the total O&S cost.

D. The next section of EIGHT prints the maintenance and Operating Support Costs. Each line of output will include the WBS code, cost description, cost, and percentage of total cost. There are three levels of output data. Level 1 is the Research and Development cost with the WBS Code of "1.000." Investment costs are the second level having a WBS code of "2.000." The level three costs are for Operating and Support with a WBS code of "3,000." The variable PER in the write statements will be the cost percentage. The cost variables of the

write statements will be described after each statement.

1.
```
    WRITE (6,74)
74 FORMAT(82X,4HCOST4X,10HPERCENTAGE)
```
These statements print the column header for the following outputs.

2.
```
    WRITE (6,82)
82 FORMAT(5X,5H1.000,5X,24HRESEARCH AND
          DEVELOPMENT)
```
These statements print the R&D cost title at WBS level 1.000.

3.
```
    WRITE (6,83)WD(1),PER(1)
83 FORMAT (5X,5H1.010,14X,23HDEVELOPMENT
           ENGINEERING19X,F20.2,4XF7.2)
```
These statements print the Development Engineering cost which was computed in LOGAM1 as F11.

4.
```
    WRITE (6,84)WD(1),PER(1)
84 FORMAT (5X,5HTOTAL56X,F20.2,4XF7.2/)
```
These statements print the total R&D cost.

5.
```
    WRITE (6,89)
89 FORMAT (5X,5H2.000,5X,15HINVESTMENT COST)
```
These statements print the Investment Cost Phase title at WBS level 2.000.

6.
```
    WRITE (6,90)WD(2),PER(2)
90 FORMAT (5X,5H2.020,9X,24HNON-RECURRING
           INVESTMENT23X,F20.2,4XF7.2)
```
These statements print the non-recurring investment which was computed in LOGAM1 as F12.

7.
```
    WRITE(6,91)WD(3),PER(3)
91 FORMAT(5X,5H2.050,9X,4HDATA43X,F20.2,4XF7.2)
```
These statements print the cost of data during the Investment Phase. WD(3) is computed in LOGAM1 as F13.

8.
```
    WRITE(6,92)WD(4),PER(4)
92 FORMAT (5X,5H2.080,9X,31HTRAINING SERVICES
          AND EQUIPMENT16XF20.2,4XF7.2)
```
These statements print Investment Training Services and Equipment costs. WD(4) is computed in LOGAM1 as F13.

9.
```
    WRITE(6,93)WD(5),PER(5)
93 FORMAT(5X,5H2.090,9X,31HINITIAL SPARES AND
          REPAIR PARTS16XF20.2,4XF7.2)
```
These statements print investment cost of Initial Spares and Repair Parts. WD(5) is computed in LOGAM1 as F15

10.
```
     WRITE(6,200) WD(6),PER(25)
```

```
200 FORMAT(5X,5H2.11 ,9X,5HOTHER,42X,F20.2,4X,F7.2)
```
These statements print Other Investment Costs. WD(6) is computed in LOGAM1 as F16, the procurement of test equipment.

11.
```
    WRITE(6,94)XB,PER(1)
94  FORMAT(5X,5HTOTAL56X,F20.2,4XF7.2/)
```
These statements print the Total Investment Cost. XB is summed in this subroutine.

12.
```
    WRITE(6,10)
10  FORMAT(5X,5H3.000,5X,26HOPERATING AND SUPPORT COST)
```
These statements print the title for the Operating and Support Costs at WBS level 3.000.

13.
```
    WRITE(6,11)
11  FORMAT(5X,5H3.010,5X,22H    MILITARY PERSONNEL
```
These statements print the Military Personnel title for costs at level 3.010 of the O&S phase.

14.
```
    WRITE(6,12)P(1),PER(6)
12  FORMAT(5X,5H3.011,5X,32H    CREW PAY AND
            ALLOWANCES19X,F20.2,4XF7.2)
```
These statements print the military personnel cost for Crew Pay and Allowances. P(1) was computed subroutine ONE.

15.
```
    WRITE(6,13)SDA(1),PER(7)
13  FORMAT(5X,5H3.012,5X,39H MAINTENANCE PAY AND
            ALLOWANCES12X1,F20.2,4XF7.2)
```
These statements print the military personnel cost for Maintenance Pay and Allowances. SDA(1) was computed in LOGAM1 as F1.

16.
```
    WRITE(6,14)P(2),PER(8)
14  FORMAT(5X,5H3.013,14X,27HINDIRECT PAY AND
            ALLOWANCES15X,F20,2,4XF7.2)
```
These statements print Indirect Pay and Allowances for military personnel. P(2) is computed in subroutine ONE.

17.
```
    WRITE(6,15)P(3),PER(9)
15  FORMAT(5X,5H3.014,14X,27HPERMANENT CHANGE OF
            STATION15X,F20.2,4XF7.2)
```
These statements print the Permanent Change of Station for military personnel. P(3) is computed in subroutine TWO.

18.
```
    WRITE(6,16)
16  FORMAT(5X,5H3.020,9X.11HCONSUMPTION)
```
These statements print a title for consumables during the O&S phase. The consumption costs are at WBS level 3.020.

19.
```
    WRITE(6,17)SDA(2),PER(10)
17  FORMAT(5X,5H3.021,14X,20HREPLENISHMENT SPARES
```

```
            22X,F20.2,4XF7.2)
```

These statements print the consumable costs for Replenishment Spares. SDA(2) was computed in LOGAM1 as F2.

20.
```
    WRITE(6,18)P(4),PER(11)
18  FORMAT(5X,5H3.022,14X,29HPETROLEUM,OIL
            AND LUBRICANTS13X,F20.2,4XF7.2)
```
These statements print the consumable costs for POL. P(4) was computed in subroutine THREE.

21.
```
    WRITE(6,19)P(5),PER(12)
19  FORMAT(5X,5H3.023,14X,36HUNIT TRAINING
            AMMUNITION AND MISSILE6X,F20.2,4XF7.2)
```
These statements print the consumable costs for Unit Training Ammuniton and Missiles. P(5) was computed in subroutine FOUR.

22.
```
    WRITE(6,21)
21  FORMAT(5X,5H3.030,9X,17HDEPOT MAINTENANCE)
```
These statements print a title for Depot Maintenance costs during O&S.

23.
```
    WRITE(6,22)SDA(3),PER(13)
22  FORMAT(5X,5H3.031,14X,5HLABOR37X,F20.2,4XF7.2)
```
These statements print the cost of Labor at Depot Maintenance. SDA(3) was computed in LOGAM1 as F3.

24.
```
    WRITE(6,23)SDA(4),PER(14)
23  FORMAT(5X,5H3.032,14X,8HMATERIEL34X,
            F20.2,4XF7.2)
```
These statements print the Depot Maintenance Materiel costs. SDA(4) was computed in LOGAM1 as F4.

25.
```
    WRITE(6,24)SDA(5),PER(15)
24  FORMAT(5X,5H3.033,14X,14HTRANSPORTATION
            28X,F20.2,4XF7.2)
```
These statements print the Depot Maintenance Transportation costs. SDA(5) was computed in LOGAM1 as F5.

26.
```
    WRITE(6,25)SDA(6),PER(16)
25  FORMAT(5X,5H3.040,9X,22HMODIFICATIONS
            MATERIAL25X,F20.2,4XF7.2)
```
These statements print the Modifications Material costs at WBS level 3.040. SDA(6) was computed in LOGAM1 as F6.

27.
```
    WRITE(6,26)
26  FORMAT(5X,5H3.050,9X,31HOTHER DIRECT
            SUPPORT OPERATIONS)
```
These statements print a title for Other Direct Support Operations cost at WBS level 3.050.

28.
```
    WRITE(6,34)SDA(7),PER(17)
```

```
34  FORMAT(5X,5H3.051,14X,27HMAINTENANCE, CIVILIAN
         LABOR15X,F20.2,4XF7.2)
```

These statements print the cost of Maintenance, Civilian Labor at Other Direct Support Operations. SDA(7) was computed in LOGAM1 as F7.

29.
```
    WRITE(6,35)P(6),PER(18)
35  FORMAT(5X,5H3.052,14X,12HOTHER DIRECT
         30X,F20.2,4XF7.2)
```

These statements print other Direct Support Operations cost. P(6) is computed in subroutine SEVEN when System Operations and support costs are to be printed. When system maintenance support costs are to be printed (called from LOGAM1) P(6) is set to SDA(10). SDA(10) is computed in LOGAM1 as F10.

30.
```
    WRITE(6,27)
27  FORMAT(5X,5H3.060,9X,27HINDIRECT
         SUPPORT OPERATIONS)
```

These statements print a title for O&S Indirect Support Operations.

31.
```
    WRITE(6,28)P(7),PER(19)
28  FORMAT(5X,5H3.061,14X,21HPERSONNEL
         REPLACEMENT21X,F20.2.4XF7.2)
```

These statements print the Personnel Replacement cost. P(7) is computed in subroutine SEVEN when System Operations and Support costs are printed. For System Maintenance Support costs P(7) is set to SDA(8) in this subroutine. SDA(8) is computed as F8 in LOGAM1.

32.
```
    WRITE(6,29)P(8)PER(20)
29  FORMAT(5X,5H3.062,14X,34HTRANSIENTS, PATIENTS
         AND PRISONERS8X,F20.2,4XF7.2).
```

These statements print the cost of Transients, Patients and Prisoners. P(8) is computed in subroutine TWO.

33.
```
    WRITE(6,30)P(9),PER(21)
30  FORMAT(5X,5H3.063,14X,35HQUARTERS, MAINTENANCE
         AND UTILITIES7X,F20.2,4XF7.2)
```

These statements print the cost of Quarters, Maintenance, and Utilities. P(9) is computed in subroutine TWO.

34.
```
    WRITE(6,31)P(10),PER(22)
31  FORMAT(5X,5H3.064,14X,15HMEDICAL
         SUPPORT27X,F20.2,4XF7.2)
```

These statements print the cost of Medical Support. P(10) is computed in subroutine TWO.

35.
```
    WRITE(6,32)P(11),PER(23)
32  FORMAT(5X,5H3.065,14X,14HOTHER
         INDIRECT28X,F20.2,4XF7.2)
```

These statements print Other Indirect Support Operations cost. P(11) is computed in subroutine SEVEN for System Operations and Support Costs. For System Maintenance Support costs P(11) is set to SDA(9) in this subroutine. SDA(9) is computed as F9 in LOGAM1.

36.
```
    WRITE(6,95)X,PER(1)
95  FORMAT(5X,5HTOTAL56X,F20.2,4XF7.2/)
```
These statements print the total cost of the O&S lifecycle phase. The total O&S cost X is summed in this subroutine.

37.
```
    WRITE(6,96)GT
96  FORMAT(5X,11HGRAND TOTAL50X,F20.2)
```
These statements print the Grand Total of all costs in the systems lifecycle. GT is summed in this subroutine.

E. The next section of subroutine EIGHT computes the difference in the LOGAM grand total cost and the sum of the costs from the PAM equations. This section of the subroutine is used to output cost deltas as a means of checking for the validity of the LOGAM costs.

1.
```
DELTA=P(12)-TAI-SDA(8)-SDA(9)-SDA(10)-WD(1)-XB
IF(J.EQ.3)DELTA=P(15)-TAI-SDA(8)-SDA(9)-SDA(10)
   -WD(1)-XB
```
These statements compute the difference in the LOGAM grand total cost and the sum of the cost from the PAM Equations. The first statement is used when outputs are from the post processor subroutine OPER. The second statement will be used when the outputs are the maintenance support costs from LOGAM1. P(15) is the same as OPERSV of LOGAM1 and P(12) is set to P(15) in subroutine SEVEN.

2.
```
    WRITE(6,160)
160 FORMAT(1H1)
    WRITE(6,150)
150 FORMAT(1H ,50X*$$$$$IMPORTANT$$$$$*)
```
These statements will cause the printer to begin printing at the top of a page where the "IMPORTANT" message is printed.

3.
```
    WRITE(6,151)
151 FORMAT(1H *THE FOLLOWING SEVEN CHECK NUMBERS
           MUST BE ZERO FOR THE 1PAM COST EQS.
           FROM LOGAM TO BE CORRECT.*)
    WRITE(6,133)X1,X2,X3,X4,X5,X6,X7
133 FORMAT(1H *CHECK EQS SHOULD BE ZERO IF PAM
           EQS ARE OK*/7E16.8)
```
These statements print the PAM cost equation error checks from LOGAM1. The "X" values were discussed in section N-65 of the LOGAM1 writeup.

4.
```
    WRITE(6,159)
    WRITE(6,152)DELTA
```

```
152 FORMAT(1H ,40X*$$$$$$$$$$THE TOTAL LOGAM
              COST MINUS PAM COST EQUALS    *E16.8,*
              $$$$$$$$$*)
    WRITE(6,159)
159 FORMAT(1H //)
    WRITE(6,158)
158 FORMAT(1H *INDIVIDUAL COST CATEGORIES
              FROM LOGAM ADDED TO PAM BREAK-OUT*)
    WRITE(6,135)WD(1),WD(2),WD(3),WD(4),WD(5),
         SDA(1),SDA(2),SDA(3)
135 FORMAT(1H 8E16.8)
    WRITE(6,135)SDA(4)SDA(5),SDA(6),SDA(7),
         SDA(8),SDA(9),SDA(10),P(12)
```

These statements print the delta cost between total LOGAM and the PAM equations. The individual cost categories are also printed here. The categories WD(1) through WD(5) are computed in LOGAM1 as F11 through F15. Categories SDA(1) through SDA(10) were computed as F1 through F10.

5.
```
RETURN
END
```

These statements terminate execution of subroutine EIGHT and returns program control to LOGAM1.

FUNCTION SPOL

This function is used to modify equipment availability when there are either built in spares are equipment redundancies. SPOL is referenced from LOGAM1 after the computations for both inherent and operational availabilities are made. The availability returned to LOGAM1 is stored in SPOL. Whether the computed availability AYZE is modified by this function depends on the values input for FN and EE.

A. This section of SPOL includes the function name and it's arguments.

1. FUNCTION SPOL(AYZE,FN,EE)
   This statement transmits the input values to the function through the argument list. The value computed in this function and returned to LOGAM1 will be stored in SPOL. The input arguments are:

   AYZE Equipment availability (either inherent or operational as computed in LOGAM1).

   FN Data input for the number of identical LRUs in a materiel system.

   EE Data input for the number of materiel systems at an Equipment installation.

B. The next section of SPOL includes the logic for modifying the input availability.

1. SPOL=1.
   IF(FN.GE.EE)RETURN
   These statements will return an availability factor of 100 percent if the number of identical LRUs is equal or greater than the number of materiel systems.

2. SPOL=AYZE
   IF(EE.EQ.1.)RETURN
   These statements will return the input availability factor when there is only one materiel system at an installation.

3. SPOL=SPOL**EE
   IF(FN.LT..5)RETURN
   This statement will compute a composite availability as a product of each materiel system and return this value, if there are not any identical LRUs in the system.

4. T=SPOL
   This statement saves the composite availability factor for the FN=1 case. This factor is used later to compute the contribution that each additional FN has on the equipment availability.

5. U=1.-AYZE
   This statement computes the compliment of the input availability. This value is used in an equation later that will compute the contribution that the FN term makes to the availability.

6.
```
  I=0
1 I=I+1
  RI=I
  T=(EE-RI)*U*T/(AYZE*RI)
  SPOL=SPOL+T
  IF(RI.GE.FN)RETURN
  GO TO 1
```
   These statements will loop through the availability computations once for each FN. When the number of loops (RI) becomes equal to the number of identical LRUs (FN) a return is made to LOGAM1 with the composite availability SPOL. On each pass through the loop, the contribution (T) that each identical LRU (RI) makes on the input availability is computed and added to the previous computed composite availability (SPOL).

7. END
   This statement terminates execution of the function.

FUNCTION AB

This function is referenced many times from LOGAM1 and three times from Subroutine ONE. Its only purpose is to return a value that is either a zero or a one. The input argument X of Function AB is checked for a value that is less than or equal to zero. If X has such a value then a value of zero is returned for AB; otherwise AB is equal to one. The function AB is used in many statements in the LOGAM program as a multiplier where only positive non-zero values contribute to the value of the statement.

The code for this function is very simple and will not be paraphrased here but the code is included below.

```
FUNCTION AB(X)
AB=1.
IF(X.LE.0.)AB=0
RETURN
END
```

FUNCTION D

This function is referenced many times from LOGAM1. It's purpose is to return a positive non zero divisor to the referencing statement. When the input argument(X) is positive and non zero, the value returned as function D is the value of X. A zero or negative value of X will result in a value of D=1.0 being returned. This function prevents a division by zero in the referencing statements.

The code for this function is rather simple and will not be paraphrased here but the code is included below.

```
FUNCTION D(X)
D=X
IF(X.GT.0)RETURN
D=1.0
RETURN
END
```

SECTION 5

PROGRAM LOGAM INPUT DEFINITIONS

5.1 LOGAM Inputs

The inputs to LOGAM consists of header control cards, LRU data cards, and post processor data cards. The header cards use formatted inputs. All other inputs use NAMELIST. There are three different sets of NAMELIST data. Two of these are for the individual LRU cases and the third is for the post processor TOE inputs. A description of all the input variables will be given in the following sections.

5.1.1 Formatted Inputs - This section describes the LOGAM header control cards. These cards input information for use as output page headers, to define the cost output units, and to control accumulation of data for summarized LRU case outputs.

1. TEXT (cards 1 through 4) - Each card may contain 72 columns of information to enable the analyst to print up to 4 lines of identifying information on each page of printed output. There must be four cards of input even if some are blank.

2. ANLYIS (card 5) - This card provides information in the first 18 columns to identify some specific information of the analysis to be printed on an output page.

3. DATE (card 6) - This card provides the first 18 columns to input a date for each page of output.

4. COSTIS (card 7) - This card contains 36 positions to input a description for the units that the cost will be output; i.e., "THOUSANDS OF DOLLARS".

5. AMULT (card 7) - Card 7 also contains the cost unit in column locations 42-51.

The five inputs above are read from the same statement. This statement is described in line item D-10 of LOGAM1.

6. TLRU (card 8) - This input is used to print with page header, information that will describe common case concepts. The first 60 columns of card 8 are reserved for this input.

7. IFLAG (card 8) - This input controls the printing of summarized LRU data cases. Any value input greater than zero will bypass the summarized data print. Column 70 is reserved for this input.

8. NDLRU (card 8) - This input instructs LOGAM as to how many LRU input cases are to be summarized into a concept. For example if NDLRU=5, every fifth LRU case will be summed; i.e., LRU1+LRU6, LRU2+LRU7, etc. This input is right justified in columns 71-80.

The READ statement for the three inputs above are described in section D-11 of LOGAM1. All of the eight parameters just described are input only once per program execution. The following two statements will be input each time an LRU data case is input.

9. UNITIS (card 9) - This input describes the class and class number for the next LRU to be processed. The first 20 columns of information is used to print with each page of LRU data outputs. Section F-10 of LOGAM1 describes the READ statement for this input.

10. REMARK (card 10) - This input provides a 72 column data field that describes the next LRU data case to be input. The READ statement for this input is described in section F-11 of LOGAM1.

5.1.2 NAMELIST/L/ - This data set contains the inputs that are peculiar to a LRU and the logistics environment in which the LRU operates. The only data not included in this section are the inputs for Type V test equipment. The Type V inputs are defined in the NAMELIST/LE/ descriptions.

This set of inputs will follow immediately after the REMARK card. The first NAMELIST card starts with $L in any card column except column 1. Column 1 must be blank for all NAMELIST inputs. The last input in the data set must be terminated by a $. In the designation of the variables for sensitivity testing, the LOGAM program is structured to reference the variables by their numbered positional location in COMMON block INPUT rather than by name. The numbers associated with the following input variable list are the numbered positions for the variables in COMMON/INPUT/. The list is in alphabetical order with the few exceptions where a few variables are better described in groups. The numberical positions are not necessarily sequenced with the alphabetic order since several variables have been added as the LOGAM program has evolved. Those inputs that do not have a sequence number is not part of COMMON/INPUT/ and therefore can not be modified with SENSY inputs. Refer to section 3.3 (special options) for a detailed descriptions of the sensitivity testing procedure.

The NAMELIST/L/inputs are:

1. ARA Annual military manpower turnover fraction for field test and repair.

322. ARAD — Annual civilian manpower turnover fraction for depot test and repair.

2. AYZP — Control to specify the method for computing the initial provision quantities. It generally is input as a signed whole number as follows:

   AYZP = 1. Use LOGAM Maintenance Rules.

   AYZP = 0. Use LOGAM Supply Rates.

   AYZP = -1. Provision quantities are to be input.

   AYZP may also contain a fractional part. The absolute value of the fraction is used to control override of stock to meet specified availability. The absolute value of the fraction states the fraction of inherent availability to be achieved.

   Example

   AYZP = 1.0 Use LOGAM Maintenance Rule. No force on availability.

   AYZP = 1.9 Use LOGAM Maintenance Rule. Force to get 90% of the inherent availability.

   AYZP = 0.73 Use LOGAM Supply Rule. Force to get 73% of the inherent availability.

3. CAD — Cost in dollars per year to retain an item (LRU, module, non-standard part) in the supply system.

4. CALMAN — Cost in dollars per year for a calibration man.

5. CALPUB — Cost in dollars for technical data for calibration/Type III test equipment. (CALPUB is set to zero within the program after use.)

6. CALSET — Number of calibration/Type III test sets and teams.

7. CCAL — Cost in dollars to develop calibration/Type III test equipment. (CCAL is set to zero within the program after use.)

8. CCALP — Cost in dollars to procure a calibration/Type III test set.

9. CCALR Cost in dollars per year to support a calibration/Type III test set.

10. CCSP Cost in dollars to develop contact support/Type IV test sets. (CCSP is set to zero within the program after use.)

11. CCSPP Cost in dollars to procure a contact support/Type IV test set.

12. CCSPR Cost in dollars per year to support a contact support/Type IV test set.

13. CDDI Shipping from Depot to General Support (units as CDEO).

14. CDEO Shipping from the installation to the Direct Support Activity. Input as dollars per item per pound per trip. Used in the computation of shipping and handling charges.

18. CDIST Cost in dollars per item per pound to distribute initial provision of LRUs, modules, and parts.

20. CDOE Shipping from Direct Support to the installation (units as CDEO).

21. CDOI Shipping from Direct to General Support (units as CDEO).

17. CDIO Shipping from General to Direct Support (units as CDEO).

15. CDFD Sh pping for a one-way trip from a contractor to the government depot (units as CDEO). Applied to shipment of reprocured material.

16. CDID Shipping from General Support to Depot (units as CDEO).

19. CDMAN Cost in dollars per year for a test man at Direct Support.

22. CDPMAN Cost in dollars per year for a test man at Depot.

23. CDPRMN Cost in dollars per year for a repairman at Depot.

24. CDRMAN Cost in dollars per year of a repairman at Direct Support.

25. CEMAN Cost in dollars per year for a test man at the Equipment level.

26. CEN Cost in dollars to enter a line item into the supply system.

27. CEND Cost in dollars to develop a LRU. (CEND is set to zero within the program after use.)

28. CERMAN Cost in dollars per year for a repairman at the Equipment level.

29. CFTD Cost in dollars per square foot/month for floor space at Depot for test equipment.

30. CGMAN Cost in dollars per year for a test man at General Support.

31. CGRMAN Cost in dollars per year for a repairman at General Support.

32. CI Cost in dollars to develop Type I test equipment. (CI is set to zero within the program after use.)

33. CII Cost in dollars to develop Type II test equipment. (CII is set to zero within the program after use.)

34. CKIT Cost in dollars for a modification kit.

35. CKMD Safety stock coefficient for module stock at Depot.

36. CKME Safety stock coefficient for module stock at equipment level.

37. CKMI Safety stock coefficient for module stock at General Support.

38. CKMO Safety stock coefficient for module stock at Direct Support.

39. CKPD Safety stock coefficient for part stock at Depot.

40. CKPI Safety stock coefficient for part stock at General Support.

41. CKPO Safety stock coefficient for part stock at Direct Support

42. CKUD Safety stock coefficient for LRU stock at Depot.

43. CKUE Safety stock coefficient for LRU stock at Equipment level.

44. CKUI Safety stock coefficient for LRU stock at General

Support.

45. CKUO Safety stock coefficient for LRU stock at Direct Support.

46. CLRUPG Cost in dollars to program and provide technical data for Type I test equipment for LRU repair.

47. CMODPG Cost in dollars to program and provide technical data for Type I test equipment for module repair for each module type.

48. CMP Cost in dollars for spare or replacement module.

49. CONMAN Cost in dollars per year/per man for the contact support team.

50. CONTCT Number of contact support sets and teams.

51. CPE Nonrecurring production cost in dollars for an LRU. (CPE is set to zero within the program after use.)

52. CPI Cost in dollars to procure a Type I test set.

53. CPII Cost in dollars to procure a Type II test equipment.

54. CPP Average cost in dollars for a spare or replacement part.

55. CPUBII Cost in dollars to program and provide technical data for Type II test equipment. (CPUBII is set to zero within the program after use.)

56. CRI Cost in dollars per year for materials to support a Type I test station.

57. CRII Cost in dollars per year for material to support a Type II test station.

58. CRM Cost in dollars per module reorder action.

59. CRP Cost in dollars per part reorder action.

60. CRU Cost in dollars per LRU reorder action.

61. CSDEP Cost in dollars per cubic foot per month for material storage at Depot.

62. CSDSU Cost in dollars per cubic foot per month for material storage at Direct Support.

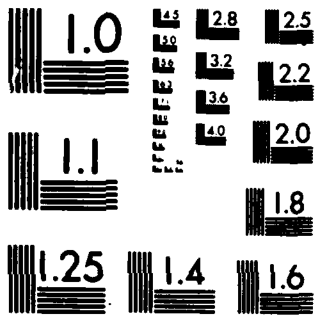
63. CSESU Cost in dollars per cubic foot per month for

AD-A136 802 LOGAM (LOGISTIC ANALYSIS MODEL) VOLUME 3
TECHNICAL/PROGRAMMER MANUAL(U) COCKERHAM (JOHN M) AND
ASSOCIATES INC HUNTSVILLE AL AUG 82

3/3

UNCLASSIFIED USAMICOM-DA-SR-D-82-2-VOL-3 F/G 9/2 NL

END
DATE
FILMED
2-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

material storage at Equipment level.

64.	CSGSU	Cost in dollars per cubic foot per month for material storage at General Support.
65.	CTCPUB	Cost in dollars to program and provide technical data for contract support/Type IV test equipment. (CTCPUB is set to zero within the program after use.)
66.	CTRA	Cost in dollars to train one man for field maintenance.
321.	CTRAD	Cost in dollars to train one man for Depot maintenance.
67.	CTRCAL	Nonrecurring cost in dollars to set up training program for calibration Type III test equipment teams.
68.	CTRI	Nonrecurring cost in dollars to set up training program for Type I test equipment.
69.	CTRII	Nonrecurring cost in dollars to set up training program for Type II test equipment.
70.	CTRSPT	Nonrecurring cost in dollars to set up training program for contact support in Type IV test equipment.
71.	CUBEM	Storage volume in cubic feet for a module.
72.	CUPEP	Storage volume in cubic feet for a part.
73.	CUBEU	Storage volume in cubic feet for an LRU.
74.	CUCE	Cost in dollars per year for equipment level manpower to provide preventive scheduled maintenance. Used in combination with SMF to model expected value manpower at the equipment level.
75.	CUP	Cost in dollars for the LRU under analysis (deployment, replacement, and provision LRUs).
76.	DAOQL	Fraction of Depot workload that is good when delivered to the field stockage point. 1-DAOQL is recycled.
77.	DD	Number of Depot level maintenance locations.
78.	DDS	Number of Depot level support points.

79. DI Number of General Support maintenance locations.

80. DIS Number of General Support supply points.

272. DTE Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from the equipment level.

273. DTO Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from Direct Support.

274. DTI Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from General Support.

81. E Failure rate per operating hour.

275. EACAL Controls posting out one time costs for calibration/Type III test channels including manpower. Only the values zero and unity are permitted.

EACAL = 0 no posting of costs.

EACAL = 1 forces the posting of costs.

EACAL is reset to zero after each use.

276. EACSP Controls posting out one time costs for contact support/Type IV test equipment and manpower. Only the values zero and unity are permitted.

EACSP = 0 no posting of costs.

EACSP = 1 forces the posting of costs.

EACSP is reset to zero after each use.

82. ED Number of deployment installations.

83. EDS Number of equipment level supply points.

84. EE The number of material systems at each deployment installation.

297. ETI Controls posting out accumulated work demands at service channels of Type I test equipment and their associated repair positions. Only the values zero and unity are permitted.

ETI = 0 no posting of costs.

ETI = 1 forces the posting of cumulative demand into the cost totals and reset the demand accumulators.

298. ETII Controls posting out accumulated work demands for service channels at Depot of Type II test equipment. Only the values zero and unity are permitted.

ETII = 0 no posting of costs.

ETII = 1 forces the posting of cumulative demand into the cost totals and resets the demand accumulators.

85. EVDM Expected value flag for test manpower at Depot.

86. EVDR Expected value flag for repair manpower at Depot.

87. EVDT Expected value flag for test equipment at Depot.

88. EVEM Expected value flag for test manpower at equipment level.

89. EVER Expected value flag for repair manpower at equipment level.

90. EVET Expected value flag for test equipment at equipment level.

91. EVIM Expected value flag for test manpower at General Support.

92. EVIR Expected value flag for repair manpower at General Support.

93. EVIT Expected value flag for test equipment at General Support.

94. EVOM Expected value flag for test manpower at Direct Support.

95. EVOR Expected value flag for repair manpower at Direct Support.

96. EVOT Expected value flag for test equipment at Direct Support.

97. FI Fraction of Type I test equipment manpower demand that is added for self-support.

98. FII Fraction of Type II test equipment manpower demand

that is added for self-support.

99. FINT Yearly interest rate. Used in the computation of present value. It is the net rate between discount rate and inflation rate. Thus, if inflation exceeds discount, FINT may be input negative. Zero input gives net cost output without discount.

100. FMD Fraction of modules that arrive at Depot that are repaired. Modules not repaired are scrapped.

101. FMI Module repair fraction at General Support.

102. FMO Module repair fraction at Direct Support.

103. FN Number of identical LRUs within a system whose failure does not detract from system availability. Used to model effect of equipment redundacy within the system.

104. FNGF Number to specify the ratio of false "no go" LRU demands to true failures.

105. FNSP Nonstandard part fraction related to the cost for supply administration.

106. FSA Field supply administration cost. Dollars per year per line item type per field supply location.

107. FTI Number of square feet of space required at Depot for Type I test equipment.

108. FTII Number of square feet of space required at Depot for Type II test equipment.

109. FTM Analogous to FTU but is for module reprocurement.

110. FTP Analogous to FTU and FTM but is for parts reprocurement.

111. FTU Time factor in weeks used in the computation of LRU Stock at Depot. FTU is the fixed time cycle associated with LRU reprocurement. Typically, this is the factory start-up time between placement of an order and delivery of the first LRU.

112. FUD LRU repair fraction at Depot.

113. FUE LRU repair fraction at equipment level.

114. FUI LRU repair fraction at General Support.

115. FUO LRU repair fraction at Direct Support.

275. GA Specifies a policy of discard at failure. There are no maintenance support activities. All failures, false "no go" indications, and attrition rate inputs result in LRU discard. Only LRUs are stocked in the supply system. There is no demand for modules or parts.

276. GB Similar to GA but here is a provision to detect false "no go's" at Direct Support and only failed and attrited LRUs are discarded. There is no demand for module or part stock. There is a demand for checkout service at Direct Support and the algebra uses Type I test equipment input data for this.

277. GC Specifies LRU repair at equipment level by removing and replacing a defective module. The defective module is discarded.

278. GD Specifies LRU repair at Direct Support by removing and replacing a defective module. The defective module is discarded.

279. GE Specifies LRU repair at General Support by removing and replacing a defective module. The defective module is discarded.

280. GF Specifies LRU repair at General Support with checkout performed at Direct Support to remove false "no go" LRUs before sending the work to General Support. LRU repair is by removal and replacement of a defective module and the defective module is discarded.

281. GG Specifies LRU repair at Depot. Defective modules are discarded.

282. GH Specifies LRU repair at Depot preceded by a checkout at Direct Support to screen false "no go's". Defective modules are discarded.

283. GI Specifies LRU repair at equipment level and module repair at Direct Support.

284. GJ Specifies LRU repair at equipment level and module repair at General Support.

285. GK Specifies LRU repair at equipment level and module repair at the Depot.

286. GL Specifies LRU and module repair at Direct Support.

287.	GM	Specifies LRU repair at Direct Support and module repair at General Support.
288.	GN	Specifies LRU repair at Direct Support and module repair at Depot.
289.	GO	Specifies checkout to catch false "no go's" at Direct Support followed by LRU and module repair at General Support.
290.	GP	Specifies checkout to catch false "no go's" at Direct Support followed by LRU repair at General Support and module repair at Depot.
291.	GQ	Specifies LRU checkout to catch false "no go's" at Direct Support followed by LRU and module repair at Depot.
292.	GR	Specifies LRU and module repair at General Support.
293.	GS	Specifies LRU repair at General Support and module repair at Depot.
294.	GT	Specifies LRU and module repair at Depot.
230. : 233.	H	An array of dimension four to specify authorized LRU supply locations.
116.	HPM	Discretionary procurement holding time in days for modules.
117.	HPP	Discretionary procurement holding time in days for parts.
118.	HPU	Discretionary procurement holding time in days for LRUs. No safety stock is applied to HPU, HPM, HPP, because it is a discretionary factor and may be waived if earlier procurement is indicated by field experience.
323.	IBG	A FLAG, which when set to 1, causes the printout of the current values of internal variables.
	IFLAG	The summation (total pages) of costs, etc. for each LRU for all theaters is suppressed. 1 Suppresses total pages. 0 Prints total pages.

INHIB An integer to control the printout of individual LRU output. Only the numbers 0 and 1 are permitted. INHIB = 0 prints the LRU output page. INHIB =1 inhibits the printout of LRU output.

299. IO An integer to control printout of the input NAMELIST data.

IO = 0 Inhibits NAMELIST printout.

IO = 1 Abbreviated NAMELIST is printed.

IO = 2 Program will print all variables in the NAMELIST.

IO = 3 Entire sequence of input data for all LRUs printed out.

324. IOPER Selects the option to add operational costs to the LOGAM output.

IOPER = 1 initiates the subroutine to compute the Operation and Support costs derived from a typical TOE structure. The O&S costs computed conform to DA PAM 11-4.

300. IS An integer to control reset functions for maintenance concept fractions, case total accumulators, availability accumulators, workload accumulators, and recall of saved input values.

IS = 1 Anticipatory control for the next LRU. All inputs used for the first LRU of the deck are recalled for use with the next LRU plus any input values keypunched for that LRU.

IS = 1 also resets availability and workload accumulators and case total accumulators.

IS ≠ 2 Resets maintenance concept fractions.

IS = 2 Retains maintenance concept fraction from one LRU to the next.

IS = 3 Neutralizes all reset actions. It must be set to 3 in the first LRU data block to assure correct accumulator function.

IPAGE An integral control for assigning the number of first page of output printout.

301. JTED An integer control used to designate the type and

location of test equipment.

JTED = 1 Permits location of Type I test equipment at the Direct Support, General Support, and Depot sites.

JTED = 2 Permits location of Type I test equipment as in JTED = 1 and Type II test equipment at Depot.

302. NA An integer to control the number of system availability modes to be tallied for the case being run.

NB An integer to control initialization of default values.

JTED = 2 Permits location of Type I test equipment as in JTED = 1 and Type II test equipment at Depot.

302. NA An integer to control the number of system availability modes to be tallied for the case being run.

NB An integer to control initialization of default values.

303. NU An integer to control printout of case totals and grand totals pages, reset the grand total accumulators and provide the means for a positive program stop.

NU = 1 Suppresses print of totals page.

NU = -1 Prints the case totals page. This value may be used at any time to examine the contents of the totals accumulators. The printout of the case totals page is not accompanied by any change in the accumulators or any other program variable.

NU = -2 Prints the case totals page as for NU = -1 and also prints a grand totals page following the case totals page. Reset of the case total accumulators is accomplished by the control IS. IS is input with the last LRU in a case deployment as IS = 1 to accomplish the reset of the case total accumulators after printout of the case totals pages.

	NU = -3	Provides the same function as NU = -2, i.e., it prints out both the case total and the grand total pages. Additionally, it resets the grand total accumulators.
	NU = -4	Provides a positive program stop; used in combination with a dummy REMARK card and a dummy UNITS card followed by a NAMELIST card with NU = -4.

119.	OD	Number of Direct Support maintenance locations.
120.	ODS	Number of Direct Support supply or stock transfer points.
234. : 237.	OL	An array of dimension four representing the operating level of supply in days for consumables at Organization, Direct, General and Depot supply points.
238. : 241.	OST	An array of dimension four representing the order and ship time in days for Organization, Direct, General, and Depot supply points.
121.	OTF	The fraction of real time that deployed equipment operates.
122.	P	Number of modules types per LRU.
123. 125. 126.	PMR, PPR, PUR	Production rates for LRUs, modules, and parts. These are not normally input because the program overrides the input if the production rates are insufficient to meet the demand and uses a value computed by the program.
124.	PP	Number of part types per LRU.
127.	QMM	The minimum reorder quantity for modules.
128.	QMP	The minimum reorder quantity for parts.
129.	QMU	The minimum reorder quantity for LRUs.
130.	QTD	Total Depot level LRU stock quantity for all DDS locations.
131.	QTE	Total organization level LRU stock quantity for all EDS locations.
132.	QTI	Total General Support level LRU stock quantity for all DIS locations.

133. QTMD Total Depot level module stock quantity for all DDS locations.

134. QTME Total organizational level module stock quantity for all EDS stock locations.

135. QTMI Total General Support level module stock quantity for all DIS locations.

136. QTMO Total Direct Support level module stock quantity for all ODS locations.

137. QTO Total Direct Support level LRU stock quanitity for all ODS locations.

138. QTPD Total Depot level part stock quantity for all DDS locations.

139. QTPI Total General Support level part stock quantity for all DIS locations.

140. QTPO Total Direct Support level part stock quantity for all ODS locations.

141. RDD Delay time in days between request for an LRU at a maintenance Depot and handling of the request by the supply point used in the computation of availability in reckoning down-time at the Depot.

142. REPEAT Number of identical LRUs in each material system.

143. RID When using LOGAM supply rules, RID is input in days and is a specification use to distinguish between the supply allowance for condemned modules and parts and the number of days of supply for LRUs and for repaired modules at the General Support level. Within the program, RID is summed with the input TDI to form the term RIDT which sets the days of supply at General Support for condemned modules and parts.

144. ROI Like RID, ROI is a specification used to distinguish between the supply allowance for condemned module and parts and the number of days of supply for LRUs and for repaired modules at the Direct Support level. Within the program, ROI is summed with the input TIO to form the term ROIT. ROIT sets the days of supply at Direct Support for condemned modules and parts.

320. REO REO is similar to ROI but in this instance, it sets the days of supply at the equipment level for condemned modules.

SENSY — An array organized in the NAMELIST format used to conduct sensitivity runs (Section 3.3).

242. : 245. SL — An array of dimension four representing the safety level days of supply for consumables at Organization, Direct, General and Depot supply points (definition of OL).

145. SMD — Module scrap fraction at Depot.

146. SME — Module scrap fraction at Organization level.

147. SMF — Scheduled maintenance fraction (CUCE definition).

148. SMI — Module scrap fraction at General Support.

149. SMO — Module scrap fraction at Direct Support.

150. SPE — Fraction for controlling the sunk portion of the prime equipment cost. Any fraction may be used for SPE, SPEV, and SPEVR.

SPE = 0 charges zero (sinks) the cost of the deployed prime equipment.

SPE = 1 charges full cost for deployed equipment.

151. SPEV — Factor to control sinking of cost of the initial provision.

SPEV = 0 no cost for the initial allowance.

SPEV = 1 charges full cost.

152. SPEVR — Factor to sink costs for consumed material

SPEVR = 0 charges zero cost.

SPEVR = 1 charges full cost.

271. STAT — The depot pipe in days between the depot and the rear-most facility shipping LRUs and modules to the depot.

153. SUD — LRU scrap fraction at Depot.

154. SUE — LRU scrap fraction at equipment level.

155. SUI — LRU scrap fraction at General Support level.

156.	SUO	LRU scrap fraction at the Direct Support level.
157.	SVE	Salvage fraction for cost of installed LRUs at the end of the life of the program.
158.	SVR	Salvage fraction for cost of consumed material (reorder stock).
159.	SVT	Salvage fraction for cost of test equipment.
160.	SVV	Salvage fraction for cost of residual inventory.
161.	TALMAN	Number of test men per calibration crew.
246. : 249.	TAT	An array of dimension four representing maintenance turn-around Time in days at Organization, Direct, General, and Depot maintenance support levels.
162.	TATE	The number of days required for stock to be obtained at the equipment level.
250. : 259.	TAYZ	An array of dimension ten to specify correspondence between model availabilities and the LRUs.
163.	TC	Mean test time in hours to checkout an LRU at any level for detection of false "no go" LRUs. Used to compute demand for test manpower.
164.	TD	Test time in hours for LRU checkout at depot. Used to compute demand for test manpower.
165.	TDI	Sums with TID to form variable TIDT which sets the number of days of supply for LRUs and for repaired modules at the General Support level. If stock of LRUs is not designated at General Support, then TIDT sums with TEOT and TOIT in computing down-time in the availability calculations (RID).
166.	TDMAN	Manpower productivity factor or number of men per test crew at Direct Support.
168.	TDPMI	Manpower productivity factor or number of men per test equipment crew at Depot (for Type I test equipment).
169.	TDPMII	Manpower productivity factor or number of men per test equipment crew at Depot (for Type II test equipment).
170.	TDPRI	Manpower productivity factor or the number of men per repair crew at Depot for Type I test equipment.

171. TDPRII Manpower productivity factor or the number of men per repair crew at Depot for Type II test equipment.

172. TDR Repair time in hours to repair an LRU. Used to compute demand at Depot.

173. TDRMAN Manpower productivity factor or number of men per repair crew at Direct Support.

174. TE Test time in hours for an LRU at equipment level. Used to compute the demand for test manpower.

175. TER Repair time in manhours for an LRU at equipment level. Used to compute the demand for repair manpower.

176. TEMAN Manpower productivity factor or number of men per test crew at equipment level.

177. TERMAN Manpower productivity factor or number of repairman per repair crew at equipment level.

178. TEO Pipelength in hours between equipment level and Direct Support when using LOGAM Supply Rules or expedited time for obtaining a spare when using LOGAM Maintenance Rules (definition of OL).

325. TENMAN The men applied to MTTR effort at equipment level. This is a multiplier of the number of eight hour shifts needed to perform the work.

179. TF Mean time in hours to test an LRU at Direct Support. It is the total time per service action in the test position and it is used to set the demand for test equipment and for test equipment men.

180. TFR Repair time in hours for an LRU at Direct Support. Used to compute demand for repair manpower.

181. TGMAN Manpower productivity factor or number of men per test crew at General Support.

182. TGRMAN Manpower productivity factor or number of repairmen per repair crew at General Support.

183. TI Test time in hours for an LRU at General Support. Used to compute demand for test manpower.

184. TID Sums with TDI to form variable TIDT which sets the number of days of supply for LRUs and for repaired modules at the General Support level. If stock or

LRUs is not designated at General Support, then TIDT sums with TEOT and TOIT in computing down-time in the availability calculations (RID).

186. TIO Sums with TOI to make the variable TOIT, TOIT states the number of days of supply at Direct Support for LRUs (repaired or condemned) and modules which will be repaired. If LRU stock is not designated at Direct, then TOIT also adds additional down-time to TEOT in the computation of availability (ROI).

187. TIR Repair time in hours of an LRU at General Support. Used to compute demand for repair manpower.

188. TMD Test time in hours for module checkout at Depot. Used to compute demand for test manpower.

190. MDR Repair time in hours for a module at Depot. Used to compute demand for repair manpower.

191. TMI Mean test in hours for module checkout at General Support. Used to compute demand for test manpower.

193. TMIR Repair time in hours for a module at General Support. Used to compute demand for repair manpower.

194. TMO Mean test time in hours for module checkout at Direct Support. Used to compute demand for test manpower.

195. TMOD, Direct,
192. TMID, General,
189. TMDD Depot
The time in hours for modification kit installation per repair crew at Direct Support, General Support, or Depot.

196. TMOR Repair time in hours for a module at Direct Support. Used to compute demand for repair manpower.

197. TOE Pipelength between Direct Support and equipment level wher using LOGAM Supply Rules, or expedited time for obtaining a spare when using LOGAM Maintenance Rules, hours (TEO).

198. TOI Sums with TIO to make the variable TOIT, TOIT states the number of days of supply at Direct Support for LRUs (repaired or condemned) and modules which will be repaired. If LRU stock is not designated at Direct, then TOIT also adds additional down-time to TEOT in the computation of availability (ROI).

199. TOMW, Direct, The mean time in hours spent in the test
185. TIMW, General, position (at Direct, General, or Depot)
167. TDMW Depot per modification per basis sequence. The program assumes that this time will be spent twice: Once before the modification is installed and once after the modification is installed.

200. TONMAN Number of men per contact support crew (Type IV test equipment).

201. TRC Down-time in hours per service demand at equipment level (equivalent to MTTR).

202. TUMD Used in concepts GN, GP, GQ, GS, and GT which call for LRU and module repair at Depot. TUMD sets the supply allowance in hours for modules at Depot to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.

203. TUMI Used in concepts GM, GO, and GR which call for LRU and module repair at General Support. TUMI sets the supply allowance in hours for modules at General Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.

204. TUMO Used for maintenance concepts GL where both LRU and module repairs are performed at Direct Support. TUMO sets the supply allowance in hours for modules at Direct Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.

205. WD The scheduled work week in hours for test equipment at Depot.

206. WDM The scheduled work week in hours for test crews at Depot.

207. WDR The scheduled work week in hours for repair crews at Depot.

208. WE Scheduled work week in hours for test equipment at Organization.

209. WEM Scheduled work week in hours for test crews at Organization.

210. WER Scheduled work week in hours for repair crews at Organization.

211. WI The scheduled work week in hours for test equipment

at General Support.

212. WIM The scheduled work week in hours for test crews at General Support.

213. WIR The scheduled work week in hours for repair crews at General Support.

214. WM The shipping weight in pounds of a module.

215. WO The scheduled work week in hours for test equipment at Direct Support.

216. WOM The scheduled work week in hours for test crews at Direct Support.

217. WOR The scheduled work week in hours for repair crews at Direct Support.

218. WP The shipping weight in pounds of a part.

219. WTKIT The shipping weight in pounds of mod kit.

220. WU The shipping weight in pounds of an LRU.

221. YAT The annual attrition fraction for LRUs. It represents an annual demand for reissue and reprocurement to replace attrited LRUs. It operates on the population of installed LRUs to determine the number to be replaced each year. Within the program, YAT is converted to an hourly attrition rate, A. A, in turn, is multiplied by OTF to get the real time rate.

222. YD The length of the development phase of the program in years. It is only used in computing present value of costs incurred during a development phase (definition for FINT).

223. YMWO The number of MWOs per year per LRU. YMWO is input as a percent per year of MWOs expected to be performed in the life cycle, i.e., if two MWOs are expected in a life cycle of 10 years, YMWO = .2.

224. YP The length of the production or acquisition phase in years. It is used in computing the present value of costs incurred during the production phase. It is also used in estimating the initial production rate which is used as a reference rate in the main program in the computation of reorder buy quantities.

225. YR The duration of the operation and maintenance portion of the program in years. Many of the cost computations for support are directly proportional to this input. It is also used in computing present value of operation and maintenance expenditures.

226. YZ Input in the dimension of years and may be positive or negative. It is used in the computation of present value of costs to change the zero print of reference at which present value is started. The program treats YD, YP, and YR as consecutive non-overlapping time intervals. Nominally, present value is computed for the end of the production phase and the start of the operation and maintenance phase. YZ shifts this point by as many years ahead of or after it. Thus, if YZ equals the negative of YP, then present value is stated at the start of the production phase. If YZ is positive, it moves the point so many years into the O&M period from it start. Shifting YZ from LRU to LRU in the input sequence of LRUs being analyzed and using sunk cost input controls can accomplish, at present value, a time phasing of program cost totals.

227. ZFL Round-off rule used in computing service channel quantities when integer round-off is invoked.

228. ZI Fraction of MWOs installed at General Support.

260. ZM An array of dimension four to specify the round-off fractions for modules at Equipment Direct, General,
:
263. and Depot supply points (ZFL).

229. ZO Fraction of MWOs installed at Direct Support.

264. : 266.	ZP	An array of dimension three to specify the round-off fractions for parts at Direct, General, and Depot supply points (ZFL).
267. : 270.	ZU	An array of dimension four to specify round-off fractions for LRUs at Equipment, Direct, General, and Depot supply points (ZFL).

5.1.3 <u>NAMELIST /LE/</u> - This data set contains the inputs for Type V test equipment. The only time this data set will be input is when ILE=1 is set in NAMELIST/L/. When ILE=1, NAMELIST/LE/ data will follow immediately after the $ entry that terminates the NAMELIST/L/ data. The rules for using /LE/ input are the same as for /L/ except the first input entry must be preceeded with $LE. The input variable descriptions are:

CV	Development cost in dollars for Type V test equipment.
CPV	Procurement cost in dollars for Type V test equipment.
CRV	Annual cost in dollars for materials to support a Type V test set.
CPUBV	Cost in dollars for technical data for Type V test equipment.
CTRV	Nonrecurring cost in dollars to set up training programs for Type V test equipment.
FE	The fraction of Type V test equipment manpower added for self-support.
ETE	Controls posting out of accumulated work demands for men and Type V test equipment. ETE=0, no posting of cost; ETE=1, posts cumulative demand into the cost total.
WMR	The work week in hours for equipment level repair men performing TRC work on major items.
WMT	The work week in hours for Type V test equipment.
RF	The fraction of TRC that is devoted to LRU remove and replace time excluding fault isolate and retest time.
ETEI	Expected value flag for test equipment on major items at the equipment level.

EREL Expected value flag for test and repair men on major items at the equipment level.

ILE Controls the input and output of NAMELIST/LE/ data. Setting ILE=1 in NAMELIST/L/ will result in reading /LE/ inputs. ILE can be turned off in /LE/ to prevent printing the NAMELIST/LE/ inputs.

5.2 Post Processor Inputs

5.2.1 NAMELIST/TOE/ - This data set contains the inputs for the Table of Organization and Equipment (TOE). /TOE/ inputs are read from the post processor driver routine OPER after the completion of all LRU cases. The NAMELIST/L/ inputs IOPER=1 and NU=-4 triggers the call to subroutine OPER. /TOE/ inputs are used to add operational costs from a typical TOE structure to the LOGAM outputs. The first data entry must be preceeded by $TOE and the last entry followed with a $. The other rules are the same as for the NAMELIST/L/ inputs.

The data is input into the T(10,200) array. The array is dimensioned to store data for 200 line items with a maximum of 10 input values per item. The meaning of each value is determined by the position of the input in the string and the values of the first (and in some cases the second) item in each 10 element field. The possible inputs for the first two values and a general description of the use of the associated data field is shown on Table 3. The first value dictates which post processor routine is called to evaluate the inputs in that 10 element data field. For example a value of 3 will cause a reference to subroutine THREE from Subroutine OPER to compute Equipment/Fuel usage costs. The second value (when applicable) determines the section of logic within a post-processor routine that will be used in the evaluation.

Table 4 includes a list of the possible input values and the mnemonics for the 10 element input fields. The mnemonics are not used in the NAMELIST inputs but are used as a reference only. In the fields where there are no mnemonics, no inputs are required. Reference Table 5 for a description of the mnemonics in Table 4.

TABLE 3

DESCRIPTION OF POSSIBLE LINE ITEMS

FIELD VALUES		DATA EVALUATED
First	Second	
1	N/A	Personnel data from the TOE
2	1	Personnel cost multipliers, list 1
2	2	Personnel cost multipliers, list 2
2	3	Personnel cost multipliers, list 3
3	N/A	Equipment/Fuel usage data
4	N/A	Ammunition usage data
5	N/A	Instrumentation Missiles
6	1	Arty/Ord inputs
6	2	Follow on training inputs
7	N/A	Signals the end of the inputs to the post-processor and the beginning of computation.
8	N/A	Program stop

TABLE 4
FIELD INPUTS AND MNEMONICS

FIELD NOS.	I	II	III	IV	V	VI	VII	VIII	IX	X
TYPE OF INPUT										
PERSONNEL COUNT	1.	QTY	OFF/EM	GRADE#	OH	DED.	CREW	SUPT	MAINT	$/YR
PERS MULT A	2.	1.	EPCSR	EPCSC	CAOD	ARCEP	CRCEP	AROEP	CROEP	AOIC
PERS MULT B	2.	2.	ARCOO	CRCOO	AROO	CROO	TPPFO	TPPFE	ACQMHO	ACQMUE
PER MULT C	2.	3.	OPCSR	OPCSC	AMSC					
FUEL USE	3.	N/A	LIN#	AULIN	RFU	CF	FOL	N/A	N/A	QLIN
AMMUNITION	4.	CATAM	CAIAM	CARSUA	CAAPLA					
INSTRUMENT MISSILES	5.	*	*	*	*	*	*	*	*	*
ARTY/ORD	6.	1.	CTEAO	CAAPLO	CGRSAO	CRUFAO	CCSFAO			FPYAO
FOT	6.	2.	CTEFOT	CAAPLF	CGRSFO	CERUFO	CCSFOT	COPFOT		
TOTAL CONTROL	7.									
STOP PROGRAM	8.									

TABLE 5

EXPLANATION OF MNEMONICS AND USE

MNEMONIC NAME	INPUT MEANING	FIELD ENTERED	FIELD I/II IDENTIFICATION
AMSC	Average annual $/person medical support	V	2/3
ACQMUE	Average annual $/enlisted man for quarters	X	2/2
ACQMUO	$/Yr./Officer or warrant/ quarters	IX	2/2
AOIC	$/Yr. other indirect cost	X	2/1
ARCEP	Attrition rate/Yr./enlisted crew	VI	2/1
ARCOO	Attrition rate/Yr./officers crew	III	2/2
AROEP	Attrition rate/Yr./Organization enlisted	VIII	2/1
AROO	Attrition rate/Yr./Organization and overhead officers	V	2/2
AULIN	Hours/Yr. energy using line item is used. Entire card refers to a specific item of equipment.	IV	3/
CAAPLA	$/Yr. for A.P.L. data takers/BN. assoc. with firings. (Ammo.)	V	4/
CAAPLF	$/Yr. for APL data takers/BN. assoc. with follow-on training firings	IV	6/2
CAAPLO	$/Yr. for APL data takers/BN. assoc. with firings (ARTY/ORD firing).	IV	6/1
CAIAM	$/Yr. associated with ammo firings for unit training	III	4/
CARSUA	$/Yr. for range support assoc. with ammo firings	IV	4/

CATAM	$/Yr. for ammunition transport	II	4/
CCSFAO	$/Yr. for contractor support for ARTY/ORD firings	VII	6/1
CCSFOT	$/Yr. for contractor support for follow-on training firings	VII	6/2
CERUFO	$/Yr. use of eastern range for follow-on test firings	VI	6/2
CF	Cost of fuel in $/gal. Appears in same line (card or group of 10 as other inputs related to same device or vehicle)	VI	3/
CGRSAO	Cost ($) per ARTY/ORD firing for range support.	V	6/1
CGRSFO	Cost ($) per FOT firing for range support.	V	6/2
COPFOT	Cost ($) per FOT firing for overseas preparation of missile	VIII	6/2
CRCEP	Replacement cost for a crew enlisted men including training.	VII	2/1
CRCOO	Replacement cost per crew officer/warrant including training.	IV	2/2
CROEP	Replacement cost per organizational enlisted men including training.	IX	2/1
CROO	Replacement cost per organizational officer/WO including training	VI	2/2
CRUFAO	Cost of range use per ARTY/ORD firing.	VI	6/1
CTEAO	Cost of transport per ARTY/ORD firing.	III	6/1
CTEFOT	Transportation cost per FOT firing	II	6/2
EPCSC	Permanent change of station	IV	2/1

	cost per enlisted men.		
EPCSR	Rate of enlisted permanent change of station (times/yr.)	III	2/1
CREW	Indication of assignment of individual represented by the line (card) to the crew. A "1" value means "crew", a "0" means other assignment.	VII	1/
DED	A "1" means individual(s) is dedicated, a "0" means not dedicated.	VI	1/
FOL	Fractional increase over fuel use to allow for oil and lube.	VII	3/
FPYAO	No. of ARTY/ORD firings/yr. for this organization.	X	6/1
FPYFOT	No. of FOT firings/yr. for this organization.	X	6/2
MAINT	A "1" indicates individual is assigned to maintenance function, a "0" indicates otherwise.	IX	1/
$/Yr.	Total pay and allowances for people represented by this line.	X	1/
SUPT	A "1" indicates individual is assigned to the support.	VIII	1/
QYT	The number of people represented by this line.	II	1/
OFF/EM	A "0" indicates line represents an officer or warrant officer, a "1" represents enlisted personnel.	III	1/
GRADE	A numerical (real number) representation of the grade of the people represented by the line. (.5 represents warrant officer).	IV	1/
OPCSC	Permanent change of station cost per officer/WO.	IV	2/3
OPCSR	Rate (no. per yr.) of change of	III	2/3

	station for officers/WO		
QLIN	The quantity of the TOE line item using the fuel.	X	3/
RFU	Rate (gal. per hr.) of usage of the fuel by each of the devices using the fuel and represented by the entry.	V	3/
*	This entry (entries) provide a means of specifying the instrumentation costs incurred in firing a missile. Nine entries are possible. They will be added by the program. The line represents a type of missile.	II thru X	5/
LIN#	This provides an opportunity to enter a number identifying the line item of the TOE.	III	3/
OH	A "1" indicates that the people identified by the entry are to be considered as overhead. A "0" indicates otherwise.	V	1/
CAOD	$/Yr. for other direct costs of the organization.	V	2/1
TPPFE	Transients, patients and prisoners factor for enlisted men. Used to increase cost on basis of pay and allowances. A fraction	VIII	2/2
TPPFO	Transients, patients and prisoners factor for officers.	VII	2/2

SECTION 6

PROGRAM LOGAM OUTPUT DEFINITIONS

6.1 Individual LRU Outputs

The following output descriptions are for the individual LRU printouts from the main program (LOGAM). These variables are printed when the input variable INHIB=0 or if NU is less than zero. The write statements for this output are described in part "R" of section 4.0 of the LOGAM program code descriptions. The reference numbers for the variables match the numbers marked on the sample output page in section 8.1.

1. IPAGE — Output page number that is printed from subroutine PAGE after a call from LOGAM.
2. TEXT — Four cards of input that describe generally the Logistics scenario.
3. UNITIS — Input description for the individual LRU class and number.
4. ANLYIS — Input description for the type of analysis.
5. REMARK — Input remark for the individual LRU.
6. DATE — Input date.

The variables above are all printed from subroutine PAGE. The following output variables are printed from LOGAM.

7. PVCGT — Present value cost total of the individual LRU.
8. PCGT — Cumulative present value cost total for LRUs analysed in a case.
9. COSTIS — Thirty six characters of input information describing the cost units.
10. AYZOS — Operational availability that includes the contribution from the inclusion of redundant equipment (LRUs).
11. AYZIS — Inherent availability.
12. CET — Development cost for prime equipment.
13. CTST — Development cost of test equipment.
14. CFT — Cost of housing test equipment.
15. CMPT — Manpower cost for test and repair personnel, including training.

16. CIVT Acquisition cost of prime equipment.

17. CROT Cost of reordering prime equipment.

18. CWHT Cost of storing prime equipment.

19. CSAT Cost of entering and maintaining line items in supply system.

20. CSHT Cost of shipping prime equipment and mod kits.

21. GCT Grand total cost.

22. QT Initial LRU stock provisions.

23. QTM Initial module stock provisions.

24. QTP Initial part stock provisions.

25. QUA Initial LRU buy quantity. Includes deployed LRUs and initial provisions.

26. QMA Initial module buy quantity.

27. QPA Initial part buy quantity.

28. QB LRU reorder buy lot size.

29. QBM Module reorder buy lot size.

30. QBP Part reorder buy lot size.

31. QC Quantity of LRUs consumed over and above the initial provisions.

32. QCM Quantity of modules consumed over and above the initial provisions.

33. QCP Quantity of parts consumed over and above the initial provisions.

34. RU Quantity of residual (salvage) LRUs.

35. RM Quantity of residual (salvage) Modules.

36. RP Quantity of residual (salvage) Parts.

37. AOY The number of manhours to test one LRU from each materiel system at one Direct Support facility.

38. SAOY Cumulative manhours at one Direct Support facility

to test all LRUs for cases currently evaluated.

39. AORY — The number of manhours to repair one LRU from each materiel system at one Direct Support facility.

40. SAORY — Cumulative manhours at one Direct Support facility to repair all LRUs for cases currently evaluated.

41. AIY — The number of manhours to test one LRU from each materiel system at one General Support facility.

42. SAIY — Cumulative manhours at one General Support facility to test all LRUs for cases currently evaluated.

43. AIRY — The number of manhours to repair one LRU from each materiel system at one General Support facility.

44. SAIRY — Cumulative manhours at one General Support facility to repair all LRUs for cases currently evaluated.

45. ADY — The number of manhours to test one LRU from each materiel system at one Depot facility.

46. SADY — Cumulative manhours at one Depot facility to test all LRUs for cases currently evaluated.

47. ADRY — The number of manhours to repair one LRU from each materiel system at one Depot facility.

48. SADRY — Cumulative manhours at one Depot facility to repair all LRUs for cases currently evaluated.

49. CAOY — Cumulative manhours at one Direct Support facility to test all LRUs for cases currently evaluated.

50. CAORY — Cumulative manhours at one Direct Support facility to repair all LRUs for cases currently evaluated.

51. CAIY — Cumulative manhours at one General Support facility to test all LRUs for cases currently evaluated.

52. CAIRY — Cumulative manhours at one General Support facility to repair all LRUs for cases currently evaluated.

53. CADY — Cumulative manhours at one Depot facility to test all LRUs for cases currently evaluated.

54. CADRY — Cumulative manhours at one Depot facility to repair all LRUs for cases currently evaluated.

55. DSU — Expected value demand for Type I test equipment at all Direct Support service channels.

56. DSUM Expected value demand on test manpower for Type I test equipment at all Direct Support service channels.

57. DSUR Expected value demand for repair manpower at all Direct Support maintenance channels.

58. GSU Expected value demand for Type I test equipment at all General Support service channels.

59. GSUM Expected value demand on test manpower for Type I test equipment at all General Support service channels.

60. GSUR Expected value demand for repair manpower at all General Support maintenance channels.

61. DEP Expected value demand for Type I test equipment at all Depot service channels.

62. DEPM Expected value demand on test manpower for Type I test equipment at all Depot service channels.

63. DEPR Expected value demand for repair manpower at all Depot maintenance channels.

64. DSUY Test manhours at one Direct Support service channel including the demand for Type I test equipment self support.

65. GSUY Test manhours at one General Support service channel including the demand for Type I test equipment self support.

66. DEPY Test manhours at one Depot service channel including the demand for Type I test equipment self support.

67. DEPAIE Expected value demand for Type II test equipment at all Depot service channels.

68. DEPAIM Expected value demand for test manpower at all Depot service channels with Type II test equipment.

69. DEPAR Expected value demand for repair manpower at all Depot service channels with Type II test equipment.

70. DEPAIY Test manhours at one Depot service channel with Type II test equipment. Includes the demand for self support.

71. EPVGCT Expected value cost total.

72. SEPV Cumulative expected value cost total for all LRU cases currently evaluated.

73. ECMPT Expected value manpower cost.

74. DELTA The difference between dedicated cost value and expected cost value.

75. PDELTA The present value for DELTA.

76. QTE Initial provision quantities for LRUs at the Equipment level.

77. QTO Initial provision quantities for LRUs at the Direct Support level.

78. QTI Initial provision quantities for LRUs at the General Support level.

79. QTD Initial provision quantities for LRUs at the Depot level.

80. QTMO Initial provision quantities for Modules at the Direct Support level.

81. QTMI Initial provision quantities for Modules at the General Support level.

82. QTMD Initial provision quantities for Modules at the Depot level.

83. QTPO Initial provision quantities for Parts at the Direct Support level.

84. QTPI Initial provision quantities for Parts at the General Support level.

85. QTPD Initial provision quantities for Parts at the Depot level.

The quantities listed above include only the provisions for unique LRUs in a materiel system. The REPEAT factor was not used in their evaluation, but REPEAT was used in costing the provisions which follow.

86. CQTE Cost of initial LRU provisions at Equipment level.

87. CQTO Cost of initial LRU provisions at Direct Support level.

88. CQTI Cost of initial LRU provisions at General Support

level.

89. CQTD Cost of initial LRU provisions at Depot level.

90. CQTT Total cost of initial LRU provisioning.

91. CRUT Salvage value of the initial LRU provisions that are unconsumed at end of O&M.

92. CQTMO Cost of initial Module provisions at Direct Support level.

93. CQTMI Cost of initial Module provisions at General Support level.

94. CQTMD Cost of initial Module provisions at Depot level.

95. CQTPT Total cost of initial Module provisioning.

96. CRMT Salvage value of the initial Module provisions that are unconsumed at end of O&M.

97. CQTPO Cost of initial Parts provisions at Direct Support level.

98. CQTPI Cost of initial Parts provisions at General Support level.

99. CQTPD Cost of initial parts provisions at Depot level.

100. CQTPT Total cost of initial parts provisioning.

101. CRPT Salvage value of the initial Parts provisions that are unconsumed at end of O&M.

6.2 Summarized LRU Case Outputs

These outputs use the same format statements used in the individual LRU outputs. The outputs printed here were summarized into the "C" array in LOGAM. The order of individual LRU cases that make up a summarized case total is determined by the NDLRU input parameter from LOGAM. As an example, if NDLRU=11, then the 1st and 12th, 2nd and 13th, etc., LRU cases are summed for the outputs here. The input flag IFLAG must be equal to zero for the values to be summarized or printed. The numbers assigned to the variables below correspond to the numbers marked on the sample output page of section 8.2 and also correspond to the numbers for the descriptions in section 6.1.

1. IPAGE Output page number that is printed from subroutine PAGE after a call from LOGAM.

2. TEXT Four cards of input that describe generally the Logistics scenario.

3. UNS Input description for the individual LRU class and number.

4. ANLYIS Input description for the type of analysis.

5. TLRU Up to 60 characters of input data to label output for summarized LRU's.

6. DATE Input date.

7. C(1) Summed values of PVGCT

8. C(2) Summed values of PCGT

9. COSTIS Same as for individual outputs.

10. Not used.

11. Not used.

12. C(3) Summed values of CET

13. C(4) Summed values of CTST

14. C(5) Summed values of CFT

15. C(6) Summed values of CMFT

16. C(7) Summed values of CIVT

17. C(8) Summed values of CROT

18. C(9) Summed values of CWHT

19. C(10) Summed values of CSAT

20. C(11) Summed values of CSHT

21. C(12) Summed values of GCT

22. C(13) Summed values of QT

23. C(14) Summed values of QTM

24. C(15) Summed values of QTP

25. C(16) Summed values of QUA

26. C(17) Summed values of QMA

27. C(18) Summed values of QPA

28. Not used.

29. Not used.

30. Not used.

31. C(19) Summed values of QC

32. C(20) Summed values of QCM

33. C(21) Summed values of QCP

34. C(22) Summed values of RU

35. C(23) Summed values of RM

36. C(24) Summed values of RP

37. C(25) Summed values of AOY

38. C(26) Summed values of SAOY

39. C(27) Summed values of AORY

40. C(28) Summed values of SAORY

41. C(29) Summed values of AIY

42. C(30) Summed values of SAIY

43. C(31) Summed values of AIRY

44. C(32) Summed values of SAIRY

45. C(33) Summed values of ADY

46. C(34) Summed values of SADY

47. C(35) Summed values of ADRY

48. C(36) Summed values of SADRY

49. C(37) Summed values of CAOY

50. C(38) Summed values of CAORY

51. C(39) Summed values of CAIY

52. C(40) Summed values of CAIRY

53. C(41) Summed values of CADY

54. C(42) Summed values of CADRY

55. C(43) Summed values of DSU

56. C(44) Summed values of DSUM

57. C(45) Summed values of DSUR

58. C(46) Summed values of GSU

59. C(47) Summed values of GSUM

60. C(48) Summed values of GSUR

61. C(49) Summed values of DEP

62. C(50) Summed values of DEPM

63. C(51) Summed values of DEPR

64. C(52) Summed values of DSUY

65. C(53) Summed values of GSUY

66. C(54) Summed values of DEPY

67. C(55) Summed values of DEPAIE

68. C(56) Summed values of DEPAIM

69. C(57) Summed values of DEPAR

70. C(58) Summed values of DEPAIY

71. C(59) Summed values of EPVGCT

72. C(60) Summed values of SEPV

73. C(61) Summed values of ECMPT

74. C(62) Summed values of DELTA

75. C(63) Summed values of PDELTA

76. C(64) Summed values of QTE

77. C(65) Summed values of QTO

78. C(66) Summed values of QTI

79. C(67) Summed values of QTD

80. C(68) Summed values of QTMO

81. C(69) Summed values of QTMI

82. C(70) Summed values of QTMD

83. C(71) Summed values of QTPO

84. C(72) Summed values for QTPI

85. C(73) Summed values for QTPD

86. C(74) Summed values for CQTE

87. C(75) Summed values for CQTO

88. C(76) Summed values for CQTI

89. C(77) Summed values for CQTD

90. C(78) Summed values for CQTT

91. C(79) Summed values for CRUT

92. C(80) Summed values for CQTMO

93. C(81) Summed values for CQTMI

94. C(82) Summed values for CQTMD

95. C(83) Summed values for CQTMT

96. C(84) Summed values for CRMT

97. C(85) Summed values for CQTPO

98. C(86) Summed values for CQTPI

99. C(87) Summed values for CQTPD

100. C(88) Summed values for CQTPT

101. C(89) Summed values for CRPT

6.3 Supplementary LRU Outputs

This output is printed from Subroutine SUPI. SUPI is referenced from LOGAM after LRU outputs are printed but only if maintenance policies (GC, GI, GJ, GK) at the equipment level are in force. SUPI is referenced twice; once for the individual LRU outputs and once for the LRU case totals. The page header outputs are the same as for the primary LRU output page.

6.3.1 <u>Individual LRU Supplementary Outputs</u> - These outputs are printed after the primary individual LRU outputs of section 6.1 are printed. The numbers assigned to the outputs below match the numbers marked on the sample output page of section 8.3.1.

1. C(106) Same as AEY in LOGAM. The manhours required to test and checkout one LRU removal from each materiel system at the Equipment level.

2. C(107) Same as BSAEY in LOGAM. The accumulated test manhours of all LRU removals at one Equipment level for the LRU cases currently evaluated.

3. C(108) Same as AERY in LOGAM. The manhours to repair one LRU from each materiel system at one Equipment level.

4. C(109) Same as BSAERY in LOGAM. The accumulated repair manhours of all LRU removals at one Equipment level for the LRU cases currently evaluated.

5. C(110) Same as BCAEY in LOGAM and is also the same as C(107) above.

6. C(111) Same as BCAERY in LOGAM and is also the same as C(108) above.

7. C(112) Same as ESU in LOGAM. Expected value demand for Type I test equipment at all Equipment levels.

8. C(113) Same as ESUM in LOGAM. Expected value demand of test manpower for type I test equipment at all Equipment level service channels.

9. C(114) Same as ESUR in LOGAM. Expected value demand for repair manpower at all Equipment levels.

10. C(115) Same as ESUY in LOGAM. Expected value demand for test manhours at one Equipment service channel. Includes the manhours added for self support of Type I test equipment.

11. C(116) Same as PQTME in LOGAM. The initial provision quantities of Modules at equipment level for unique LRUs only. Does not include the REPEAT factor.

12. C(117) Same as CQTME in LOGAM. The cost of initial provision quantities of Modules at the Equipment level. The module cost includes the REPEAT factor.

6.3.2 Supplementary Outputs for Summarized LRU Cases - The outputs listed here are the accumulated case totals for the variables described in section 6.3.1 above. The numbers assigned to each variable match the numbers marked on the sample output page of section 8.3.2. These outputs are printed after the summarized LRU case outputs are printed.

1. C(90) Case total for AEY
2. C(91) Case total for BSAEY
3. C(92) Case total for AERY
4. C(93) Case total for BSAERY
5. C(94) Case total for BCAEY
6. C(95) Case total for BCAERY
7. C(96) Case total for ESU
8. C(97) Case total for ESUM
9. C(98) Case total for ESUR
10. C(99) Case total for ESUY
11. C(100) Case total for PQTME
12. C(101) Case total for CQTME

6.4 Cost Totals Output

Cost total outputs are printed from LOGAM for CASE TOTALS and GRAND TOTALS. The same WRITE and FORMAT statements are used for both printouts. The user controls these outputs with the NU input flag. NU = -1 will print the values accumulated from the individual LRU cases as a CASE TOTAL. With NU = -2 or -3 a case total is printed and also the accumulated values for all LRU cases evaluated will be printed as a GRAND TOTAL. When NU = -2 the grand total accumulator is not reinitialized unless IS = 1 is input. NU = -3 will reinitialize the grand total accumulators and any additional individual LRU case input will start a new grand total. The NU printout selection is

entered along with the input data for the last individual LRU case that makes up a case total.

6.4.1 Case Totals - The case total printout is shown on the sample output page of section 8.4.1. This page is printed when the user inputs NU=-1 with the last individual LRU case in a case total concept. The numbers assigned to the output variable definitions correspond to the numbers marked on the sample page. The page header information for this output is the same as was discussed for the "INDIVIDUAL LRU OUTPUT" of section 6.1. The parameters enclosed in parenthesis are those evaluated for an individual LRU case. The sample case for this output is shown in section 8.4.1 of this document.

1. COSTIS Cost output units.
2. CCET Development and Procurement cost of prime (installed) equipment less the salvage value (CET).
3. CCTS Development, Procurement, Maintenance, and Software cost of test equipment less salvage value (CTST).
4. CCTSR Test equipment support cost (CTSR).
5. CCF Test equipment housing cost (CFT).
6. CCF Depot space/utilities cost (same as 5 above).
7. CCM Manpower cost for test and repair personnel including training (CMPT).
8. CCMF Field maintenance cost.
9. CCMD Depot maintenance cost.
10. CCMFD Total maintenance cost.
11. CTRF Field training cost for test and repair personnel.
12. CTRDEP Depot training cost for test and repair personnne[.
13. CTR Total training cost for test and repair personnel.
14. CIV Cost of initial provisions and consumed material during the O&M phase less the salvage value (CIVT).
15. CIVREC Cost of consumed materials (CIVR).
16. CRT Cost of reordering material (CROR).
17. CRT Cost of reordering material (CROR). Same as 16.

18. CWH cost of storage for units, modules, and parts (CWHT).

19. CWH Cost of storage for units, modules, and parts (CWHT). Same as 18.

20. CSA Cost of entering and maintaining line items in supply system (CSAT).

21. CSAREC Cost to maintain (inventory management) supply in the system (CSAR).

22. CSH Cost of shipping units, modules, parts, and mod kit (CSHT).

23. CSH Cost of shipping prime equipment and mod kits (CSHT). Same as 22.

24. CGT Case total cost (GCT).

25. CTREC Total recurring costs.

26. PCD Total development cost. Includes the cost to develop prime and test equipment plus test equipment software (PVCD).

27. CQTU Cost of initial LRU provisions (CQTT).

28. PCP Total acquisition cost. Includes procurement of prime equipment, initial provisions, nonrecurring training costs, and the cost to enter items in inventory (PVCP).

29. CQTM Cost of initial Module provisions (CQTMT).

30. PCR Total cost of Operation and Maintenance of prime equipment, initial provisions, and test equipment (PVCR).

31. CQTP Cost of initial Parts provisions (CQTPT).

32. PCS The end of life salvage value (PVCS).

33. CQTUMP Total initial provisioning cost.

34. PCGT Present value case total cost (PVGCT).

35. SEMPT Expected value cost for maintenance manpower (ECMPT).

36. SEPC Expected value case total cost (EPCGT).

37. SPCR Present value Operation and Support cost.

38. SEPV Present value grand total cost (EPVCGT).

39. SDEL The expected value cost difference (DELTA).

40. SPDEL The present value cost difference (PDELTA).

6.4.2 Grand Total Output - The grand total printout is shown on the sample output page of section 8.4.2. This page was printed when the user input NU = -2 or -3 with the last individual LRU case in a case total concept.

The grand totals are accumulated in LOGAM into the array SUM. At the time to print the grand totals the contents of SUM are stored into the CUM array which is equivalenced to the variable names listed in section 6.4.1 above. Therefore, printing this list of names again will give the grand total outputs. Since the variable list was described in section 6.4.1 they will not be included here.

6.5 Maintenance Outputs

Maintenance outputs are printed from LOGAM after case total outputs are printed. The outputs here are manhours and men for test and repair at the maintenance locations. The outputs are shown on the sample output page section 8.5. The numbers assigned to the output variables correspond to the numbers marked on the sample page.

In addition to the maintenance outputs there are other sets of information printed on the page. One set is the cost deltas (X1 through X7) computed in LOGAM. The cost deltas are the differences between the total cost of an item and the sum of the individual costs that make up that item. The total cost of items are often computed from a different aspect than are the individual costs. The output for the deltas are used to check for the correctness of the PAM cost equations. A second set is the individual cost categories from LOGAM that are added to the PAM break outs. System and subsystem availabilities are also printed on this output page.

1. X1 Cost of test equipment.

2. X2 Salvage value of consumed materials.

3. X3 Salvage value of consumed materials including mod kits.

4. X4 Cost for initial provisions.

5. X5 Salvage value of unconsumed stock.

6. X6 Test equipment personnel related costs.

7. X7 Cost of consumed materials.

The outputs above are the difference checks for PAM equation correctness. These outputs are computed in LOGAM and printed from Subroutine EIGHT.

8. DELTA Difference check for the grand total cost. DELTA is computed and printed from Subroutine EIGHT.

9. WD(1) Development cost (CED+CTSD).

10. WD(2) Non Recurring Investment (CEP+CEV+CTSV).

11. WD(3) Cost of Data (CTSOFT).

12. WD(4) Cost of training services and equipment (CMPPY).

13. WD(5) Cost of initial spares and repair parts (CIVP).

14. SDA(1) Pay and allowance for military personnel (PA).

15. SDA(2) Cost of consumed and support material at Field (RESPF).

16. SDA(3) Cost of personnel at Depot (DML).

17. SDA(4) Cost of consumed and support material at Field (RESPD).

18. SDA(5) Cost of shipping prime equipment and mod kits to Depot (CSHTD).

19. SDA(6) Cost of mod kits (DMM).

20. SDA(7) Civilian maintenance labor cost (CLS).

21. SDA(8) Cost of training test and repair personnel (U12).

22. SDA(9) Cost of housing test equipment and shipping mod kits to Field (CSHTF + CFT).

23. SDA(10) Cost of maintaining items in supply system (U17).

The outputs above are case totals that were accumulated in LOGAM. SDA(1) through SDA(10) are equivalent to F1 through F10 and WD(1) through WD(5) are equivalent to F11 through F15. The variables enclosed in paranthese[are the individual LRU cost elements evaluated in LOGAM. These outputs are printed from Subroutine Eight.

24. P(12) This parameter is set equal to zero for all case total outputs. When a grand total output is being printed during the post processor phase, P(12) is equal to the case total for Operating and Support costs (OPERSV).

25. CAYZ Operational availabilities.

26. CAYZI Inherent availabilities.

The last two outputs above are evaluated in LOGAM. The number (NA) of values printed per array name is controle[by the number of system availability modes input by the user. The first value of each array is the system availability and the subsequent values are subsystem availabilities. The control for system availability computations is determined by the TAYZ input array. Refer to section 3.2 for a detailed description of the TAYZ inputs.

The following outputs are the case totals that were accumulated from the individual LRU cases in the main program (LOGAM).

27. WPD(1,1) Test manhours/year at all equipment installations.

28. WPD(2,1) Test manhours/year at all equipment maintenance facilities.

29. WPD(3,1) Test manhours/year at all Direct Support maintenance facilities.

30. WPD(4,1) Test manhours/year at all General Support maintenance facilities.

31. WPD(5,1) Test manhours/year at all Depot maintenance facilities.

32. WPD(1,2) Repair manhours/year at all Equipment installations.

33. WPD(2,2) Repair manhours/year at all Equipment maintenance facilities.

34. WPD(3,2) Repair manhours/year at all Direct Support maintenance facilities.

35. WPD(4,2) Repair manhours/year at all General Support maintenance facilities.

36. WPD(5,2) Repair manhours/year at all Depot maintenance facilities.

37. PERS(1,1) Test manpower at all equipment installations.

38. PERS(2,1) Test manpower at all Equipment maintenance facilities.

39. PERS(3,1) Test manpower at all Direct Support maintenance facilities.

40. PERS(4,1) Test manpower at all General Support maintenance facilities.

41. PERS(5,1) Test manpower at all Depot maintenance facilities.

42. PERS(1,2) Repair manpower at all equipment installations.

43. PERS(2,2) Repair manpower at all equipment maintenance facilities.

44. PERS(3,2) Repair manpower at all Direct Support maintenance facilities.

45. PERS(4,2) Repair manpower at all General Support maintenance facilities.

46. PERS(5,2) Repair manpower at all Depot maintenance facilities.

47. WPDL(1,1) Test manhours/year per equipment installation.

48. WPDL(2,1) Test manhours/year per equipment maintenance facility.

49. WPDL(3,1) Test manhours/year per Direct Support maintenance facility.

50. WPDL(4,1) Test manhours/year per General Support maintenance facility.

51. WPDL(5,1) Test manhours/year per Depot maintenance facility.

52. WPDL(1,2) Repair manhours/year per equipment installation.

53. WPDL(2,2) Repair manhours/year per equipment maintenance facility.

54. WPDL(3,2) Repair manhours/year per Direct Support maintenance facility.

55. WPDL(4,2) Repair manhours/year per General Support maintenance facility.

56. WPDL(5,2) Repair manhours/year per Depot maintenance facility.

57. PERL(1,1) Test manpower per equipment installation.

58. PERL(2,1) Test manpower per Equipment maintenance facility.

59. PERL(3,1) Test manpower per Direct Support maintenance facility.

60. PERL(4,1) Test manpower per General Support maintenance facility.

61. PERL(5,1) Test manpower per Depot maintenance facility.

62. PERL(1,2) Repair manpower per equipment installation.

63. PERL(2,2) Repair manpower per Equipment maintenance facility.

64. PERL(3,2) Repair manpower per Direct Support maintenance facility.

65. PERL(4,2) Repair manpower per General Support maintenance facility.

66. PERL(5,2) Repair manpower per Depot maintenance facility.

6.6 System Support Costs Output

There are two levels of system support costs that are printed. Subroutine EIGHT is called from LOGAM to print System Maintenance Support Costs each time a case total or a grand total output is printed. The same output statements are used to print System Operating and Support Costs when Subroutine EIGHT is later referenced from the post processor subroutine OPER. The O&S costs include the maintenance costs from LOGAM and the operating costs evaluated from the TOE inputs to the post processor.

6.6.1 System Maintenance Support Costs - The numbers assigned to the output descriptions correspond to the number marked on sample output page of section 8.6.1. Only those outputs that are evaluated in LOGAM will be listed in this section. The variables enclosed in parentheses are the names used in LOGAM to compute the cost at the individual LRU case level.

1. WD(1) Development Engineering costs (CED+CTSD).

2.	PER(1)	Percent of R&D cost for WD(1)
3.	WD(1)	Total research and Engineering cost.
4.	PER(1)	Percent of R&D cost for ED(1) (100%).
5.	WD(2)	Non recurring investment cost (CEP + CEV + CTSV).
6.	PER(2)	Percent of Total Investment cost for WD(1) (100%).
7.	WD(3)	Cost of Data (CTSOFT).
8.	PER(3)	Percent of Total Investment cost for WD(3).
9.	WD(4)	Cost of Training Services and Equipment (CMPPY).
10.	PER(4)	Percent of Total Investment cost for WD(4).
11.	WD(5)	Cost of Initial Spares and Repair Parts (CIVP).
12.	PER(5)	Percent of Total Investment cost for WD(5).
13.	WD(6)	Other Investment costs (CTSD).
14.	PER(6)	Percent of Total Investment cost for WD(6).
15.	XB	Total Investment cost.
16.	PER(1)	Percent of Total Investment cost for XB (100%).
19.	SDA(1)	Maintenance pay and allowances (PA).
20.	PER(7)	Percent of O&S cost for SDA(1).
25.	SDA(2)	Cost of Replenishment spares (REPSF).
26.	PER(10)	Percent of O&S cost for SDA(2).
31.	SDA(3)	Cost of Labor (DML).
32.	PER(13)	Percent of O&S cost for SDA(3).
33.	SDA(4)	Cost of materials (RFPSD).
34.	PER(14)	Percent of O&S cost for SDA(4).
35.	SDA(5)	Cost of Transportation (CSHTD).
36.	PER(15)	Percent of O&S cost for SDA(5).
37.	SDA(6)	Cost of modification materials (DMM).

38. PER(16) Percent of O&S cost for SDA(6).

39. SDA(7) Cost of maintenance civilian labor (CLS).

40. PER(17) Percent of O&S Cost of SDA(7).

41. P(6) Other Direct O&S costs (U17). Same as SDA(10).

42. PER(18) Percent of O&S cost for P(6).

43. P(7) Personnel Replacement cost (U12). Same as SDA(8).

44. PER(19) Percent of O&S cost for P(7).

51. P(11) Other Indirect cost (CSHTF + CFT). Same as SDA(9).

52. PER(23) Percent of O&S Cost for P(11).

53. X Total Operating and Support Cost.

54. PER(1) Percent of O&S cost for X (100%).

55. GT Grand Total cost.

6.6.2 System Operations and Support Costs - These outputs are printed when Subroutine EIGHT is called from the post processor Subroutine OPER. Only those outputs that are evaluated in the post processor will be listed here. All other outputs are the LOGAM grand totals described in section 6.6.1. The numbers assigned to the output list correspond to the numbers marked on the sample output page of section 8.6.2.

17. P(1) Crew pay and allowances.

18. PER(6) Percent of O&S cost for P(1).

21. P(2) Indirect pay and allowance.

22. PER(8) Percent of O&S cost for P(2).

23. P(3) Permanent change of Station costs.

24. PER(9) Percent of O&S for P(3).

27. P(4) Cost of petroleum, oil and lubricants.

28. PER(1) Percent of O&S for P(4).

29. P(5) Cost of Unit Training Ammunition and Missiles.

30. PER(12) Percent of O&S for P(5).

41. P(6) Other Direct Costs (includes SDA(10) from LOGAM).

42. PER(18) Percent of O&S for P(6).

43. P(7) Cost of Personnel Replacement (includes SDA(8) from LOGAM).

44. PER(19) Percent of O&S cost for P(7).

45. P(8) Cost of Transients, patients and prisioners.

46. PER(20) Percent of O&S cost for P(8).

47. PER(21) Cost of quarters, maintenance and utilities.

48. PER(21) Percent of O&S cost for P(9).

49. P(10) Cost of Medical Support.

50. PER(22) Percent of O&S cost for P(10).

51. P(11) Cost of other Indirect O&S ccsts (includes SDA(9) from LOGAM).

52. PER(23) Percent of O&S cost for P(11).

6.7 Sensitivity Outputs

The sensitivity output uses the same program formats as described earlier in section 6.1. An additional line of print is included with the Individual LRU output pages that describe the sensitivity analysis being performed. The Individual LRU outputs can be inhibited by inputting INHIB = 1. In this instance only the description of the sensitivity will be output per individual LRU case except for the last case of a concept. The last individual LRU case in a concept will always be printed. Section 6.7.1 describes the sensitivity case where INHIB = 1 and section 6.7.2 describes the Individual LRU case output. Case totals, grand totals, and all other output forms are printed as described earlier for the baseline case.

6.7.1 Sensitivity with Inhibited Print - The additional print given when a sensitivity analysis is being performed is shown on the sample output pages of sections 8.7.1 and 8.7.2. The numbers assigned to the following descriptions correspond to the numbers marked on the sample outputs.

1. UNITIS This output normally describes the class and class number of an individual LRU case. It can be used when inputting SENSY data to describe the

sensitivity analysis. Subroutine PAGE is called when INHIB=1 to print the UNITIS information along with the other page header data. Line item 9 of section 5.1.1 describes the UNITIS input.

2. REMARK — A 72 column input field that describes an individual LRU case; in this instance it is used to describe a sensitivity analysis. REMARK is printed from subroutine PAGE. Line item 10 of section 5.1.1 describes this input.

3. KPASS — The sensitivity pass number. An LRU input value can be varied one or more times using the SENSY input array. KPASS will count each variation to the input value. According to the "REMARK" on sample output 8.7.1, the failure rate (E) is to be varied 2 and 3 times the baseline input. KPASS is equal to 1 on the first pass through the LRU cases with "E" twice the baseline value and is equal 2 when the LRU cases are recycled with "E" three times the baseline. This and the following variables are printed in subroutine SENSIT.

4. NRU — The individual LRU case number. The case number is the order that the LRU case occure[in the NAMELIST/L/ data decks.

5. NVAR — The modified variables position within the list of variables in COMMON/INPUT/. In the sample output of section 8.7.1 the variable modified is E (failure rate) which is in position 81 of COMMON/INPUT/. Refer to input section 5.1.2 for a list of the variable that can be modified by SENSY and their positions.

6. VALUE — The modified value for the variable stored in position NVAR of COMMON/INPUT/. The sample output of section 8.7.1 shows only one variable per case being modified. There can be several different variable modifications per case if so desired.

The line of print following the sensitivity print on the section 8.7.1 output page is debugging output to check the correctness of cost computations. These outputs are the X1 through X7 values discussed in section 6.5.

The following section discusses the sensitivity outputs when individual LRU case outputs are not inhibited (INHIB=0). The sample output (section 8.7.2) associated with this discussion is also output for the last LRU in a case total even when the output is inhibited.

6.7.2 Sensitivity with Uninhibited Output - The outputs when a sensitivity analysis is being performed and INHIB = 0 are the same as discussed in sections 6.1 through 6.6 except that the sensitivity input descriptions are included on the individual LRU output page. Section 8.7.2 is a sample output page for an individual LRU with the sensitivity data. The numbers assigned to the following statements correspond to the numbers marked on the output page of section 8.7.2.

7. UNITIS This is the class and class number description that was input with the baseline case.

8. REMARK This is the 72 column description of on LRU that was input with the baseline case.

9. KPASS This variable and the ones following are printed in Subroutine SENSIT. KPASS is the same as was discussed in section 6.7.1.

10. NRU Same as in section 6.7.1.

11. NVAR Same in section 6.7.1.

12. VALUE Same as in section 6.7.1.

The additional outputs on the sample output page of section 8.7.2 are discussed in section 6.1 for Individual LRU Outputs.

SECTION 7

SAMPLE INPUTS

7.1 LOGAM Sample Input Case

```
COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
 USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
  MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
     ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.
THREE LRU CLASSES
AUG 1982
THOUSANDS OF DOLLARS                        .001
   TOTAL                                                          11
CLASS 1 LRU NO. 1
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L OTF=.0548,E=.0001,P=3.,PP=20.,TIMW=.5,TDMW=.5,RDD=30.,WU=7.5,WM=.1,
 CKUI=.85,CKUD=.85,CKMO=.85,CKMI=.85,CKMD=.85,CKPO=.85,CKPI=.85,
 WP=.05,CUBEU=.12,CUBEM=.005,CUBEP=.003,CDMAN=16600.,CDRMAN=16600.,
 CGMAN=16600.,CGRMAN=16600.,WO=100.,WI=100.,WD=100.,ED=141.,IO=0,
 CTRA=2350.,ARA=.4,CSDEP=.1,DADQL=.98,CUP=988.,CMP=500.,CPP=3.,
 YMWO=.2,TMID=1.,TMDD=1.,CDPMAN=26100.,CDPRMN=26100.,ZI=0.,GT=1.,DTO=60.,
 CEN=1077.,CAD=436.,TMD=.8,TMDR=1.3,DI=2.,TDMAN=2.,TDRMAN=2.,DTI=60.,
 TOI=17.,TIO=0.,TDI=60.,TID=30.,NU=1,IS=3,CKIT=148.,TAYZ=2*1.,8*0.,NA=4,
 FTU=64.,FTM=38.,FTP=20.,CRU=835.,CRM=835.,CRP=835.,WTKIT=1.,TUMD=336.,
 WOM=40.,WIM=40.,WDM=40.,WOR=40.,WIR=40.,WDR=40.,YD=1.,YP=1.,YR=10.,
 TUMO=0.,INHIB=0,YZ=-1.5,CFTD=1.,QMU=20.,QMM=50.,QMP=100.,CKUO=.85,
 CKPD=.85,TGMAN=2.,TGRMAN=2.,TDPMI=2.,TDPRI=2.,TDPMII=2.,TDPRII=2.,
 FI=.1,FII=.1,OD=9.,ODS=9.,FNSP=.5,EVOR=0.,EDS=141.,FNGF=.2,DIS=2.,
 AYZP=1.,RID=30.,ROI=17.,SMD=0.,SMI=0.,SUD=0.,SUI=0.,TATE=3.,TOE=2.,
 SL=15.,15.,30.,TAT=15.,30.,127.,HPM=30.,HPP=30.,HPU=30.,ZU(1)=.5,
 CKUE=.85,CKME=.85,REO=17.,OL=15.,15.,15.,30.,OST=15.,15.,15.,30.,
 DTE=6.,SL=1.5,1.5,1.5,3.,TAT=1.5,1.5,1.5,3.,
 CEMAN=16600.,CERMAN=16600.,TE=1.,TER=1.,TEMAN=2.,TERMAN=2.,
 TRC=1.,TUMI=0.,YAT=0.,H=4*1.,OL=15.,15.,30.,OST=15.,15.,30.,STAT=60.,
 TATE=30.,STAT=30.,
 TATE=60.,
 GT=0.,
 AYZP=0.,
 CDIST=.6,
 GC=1. $
CLASS 1 LRU NO. 2
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L IO=0,E=.0001,CUP=988.,CMP=500.,WP=.1,CUBEP=.005,TMD=.6,TMDR=1.1,
 WU=4.5,WM=.2,CUBEU=.15,CUBEM=.015,P=3.,PP=30.,CPP=2.5,GI=1.,GC=0.,
 $
```

7.1 <u>LOGAM Sample Input Case</u> (Continued)

```
CLASS 1 LRU NO. 3
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L E=.0005,P=2.,WP=.1,TI=.25,TIR=.5,TD=.25,TDR=.5,TMD=.5,TMDR=.9,
 WU=3.,WM=.5,CUBEU=.1,CUBEM=.01,CUBEP=.005,CUP=988.,CMP=500.,PP=20.,
 CPP=7.,GI=0.,GJ=1.,
 $
CLASS 1 LRU NO. 4
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L E=.0005,P=2.,PP=0.,TI=.25,TIR=.5,TD=.25,TDR=.5,TMD=0.,TMDR=0.,
 WU=3.,WM=.5,WP=0.,CUBEU=.1,CUBEM=.01,CUBEP=0.,CMP=450.,CPP=0.,
 CUP=741.,GT=1.,IO=1,CKIT=111.,GT=0.,GI=0.,GJ=1.,
 $
CLASS 2 LRU NO. 1
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L E=.0021,ZI=.5,PP=50.,WU=40.,WM=2.,WP=.1,CUBEU=.75,CUBEM=.02,P=15.,
 TI=2.,TD=2.,TMD=.5,TMDR=.9,TDR=2.,TIR=2.,CUBEP=.005,CUP=57730.,
 TAYZ=1.,0.,1.,7*0.,
 CMP=2080.,CPP=12.5,IO=0,CDPMAN=16600.,CDPRMN=16600.,GJ=0.,GK=1.,
 CKIT=5773.,WTKIT=10.,
 $
CLASS 2 LRU NO. 2
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L E=.0017,PP=40.,TI=1.8,TD=1.8,P=10.,CMP=1126.,CPP=18.,TIR=1.5,WU=26.,
 IBG=0,
 CKIT=1761.,
 WM=1.5,WP=.08,CUBEU=.7,CUP=17613.,TMD=.5,TMDR=.9,TDR=1.5,GS=.85,GT=.15,GK=0.,
 $
CLASS 2 LRU NO. 3
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L E=.0011,P=8.,TI=.5,TD=.5,WU=36.,WM=2.,WP=.1,CUBEU=.75,CPP=10.5,
 TMD=.4,TMDR=.8,CUP=18827.,PP=40.,CMP=1500.,GS=.85,GT=.15,CKIT=1883.,
 IBG=0,
 $
CLASS 2 LRU NO. 4
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L TI=.8,TIR=1.8,TD=.8,TDR=1.8,E=.001,TMD=.3,TMDR=.6,CUP=12250.,P=4.,
 WU=40.,WM=2.,WP=.1,CUBEU=.75,CUBEM=.02,CUBEP=.005,PP=40.,CMP=1360.,
 CPP=9.,GS=.85,GT=.15,CKIT=1225.,
 $
CLASS 2 LRU NO. 5
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L TI=1.,TIR=1.8,TD=1.,TDR=1.8,TMD=.1,TMDR=.4,WU=36.,CKIT=500.,
 CUP=5000.,P=4.,PP=40.,CMP=1000.,CPP=6.,E=.0008,IO=1,
 CI=1824000.,CPI=131500.,CRI=6000.,ETI=1.,GS=.85,GT=.15,IO=0,
 $
```

7.1 LOGAM Sample Input Case (Continued)

```
CLASS 3 LRU NO. 1
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L EE=1.,E=.001,P=12.,PP=50.,TI=.5,TIR=1.6,TD=.5,TDR=1.6,TMD=.3,
 TMDR=.6,WU=30.,WM=1.5,WP=.08,CUBEU=1.,CUBEM=.05,CUBEP=.01,CUP=27716.,
 TAYZ=1.,2*0.,7*1.,
 CPI=0.,CRI=0.,
 CMP=1610.,CPP=6.,IO=0,    ZI=.7,GS=.7,GT=.3,DI=1.,CKIT=2772.,DIS=1.,
 $
CLASS 3 LRU NO. 2
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS
 $L NU=-1,IS=1,          ETII=1.,E=.0013,P=13.,PP=40.,TI=1.,IO=2,IO=0,
 CUBEU=15.,CUBEM=.5,CUBEP=.05,WP=.5,CII=1370000.,CRII=7500.,
 CI=0.,CPI=0.,CRI=0.,TMDR=3.4,CPP=11.,CPII=264000.,
 CCSP=425000.,CCSPP=100000.,             CCSPR=1000.,EACSP=1.,
 TD=1.,TMD=.75,CMP=2500.,TDR=3.5,TIR=3.5,WM=15.,CUP=75262.,WU=150.,
 EACAL=1.,CCALP=220000.,CALSET=1.,CCALR=2000.,ETI=1.,DI=1.,
 CONTCT=10.,
 ZI=.7,GS=.7,GT=.3,CKIT=7526.,WTKIT=30.,DIS=1.,
 $
CLASS 1 LRU NO. 1
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L  DTO=30.,DTI=30.,ED=40.,EDS=40.,OD=4.,ODS=4., DI=4.,DIS=4.,TOI=0.,TIO=17.,
 TDI=30.,FTU=56.,FTM=30.,FTP=12.,OST(3)=20.,STAT=20.,CEN=451.,CAD=170.,
 GT=1.,
 CDFD=.33,CDIST=.3,
 $
CLASS 1 LRU NO. 2
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L IO=0,E=.0001,CUP=988.,CMP=500.,WP=.1,CUBEP=.005,TMD=.6,TMDR=1.1,
 WU=4.5,WM=.2,CUBEU=.15,CUBEM=.015,P=3.,PP=30.,CPP=2.5,GT=1.,
 $
CLASS 1 LRU NO. 3
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L E=.0005,P=2.,WP=.1,TI=.25,TIR=.5,TD=.25,TDR=.5,TMD=.5,TMDR=.9,
 WU=3.,WM=.5,CUBEU=.1,CUBEM=.01,CUBEP=.005,CUP=988.,CMP=500.,PP=20.,
 CPP=7.,GT=1.
 $
CLASS 1 LRU NO. 4
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L E=.0005,P=2.,PP=1.,TI=.25,TIR=.5,TD=.25,TDR=.5,TMD=0.,TMDR=0.,
 WU=3.,WM=.5,WP=0.,CUBEU=.1,CUBEM=.01,CUBEP=0.,CMP=450.,CPP=0.,
 PP=0.,
 CUP=741.,GT=1.,IO=2,CKIT=111.,IO=0,
 $
```

7.1 LOGAM Sample Input Case (Continued)

```
CLASS 2 LRU NO. 1
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L E=.0021,ZI=.5,PP=50.,WU=40.,WM=2.,WP=.1,CUBEU=.75,CUBEM=.02,P=15.,
 TI=2.,TD=2.,TMD=.5,TMDR=.9,TDR=2.,TIR=2.,CUBEP=.005,CUP=57730.,
 TAYZ=1.,0.,1.,7*0.,
 CMP=2080.,CPP=12.5,GS=.85,GT=.15,CKIT=5773.,WTKIT=10.,
 $
 CLASS 2 LRU NO. 2
 CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
  $L E=.0017,PP=40.,TI=1.8,TD=1.8,P=10.,CMP=1126.,CPP=18.,TIR=1.5,
  WU=26.,WM=1.5,WP=.08,CUBEU=.7,CUP=17613.,TMD=.5,TMDR=.9,TDR=1.5,
  GS=.85,GT=.15,IO=0,CKIT=1761,
  $
 CLASS 2 LRU NO. 3
 CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
  $L E=.0011,P=8.,TI=.5,TD=.5,WU=36.,WM=2.,WP=.1,CUBEU=.75,CPP=10.5,
  TMD=.4,TMDR=.8,CUP=18827.,PP=40.,CMP=1500.,GS=.85,GT=.15,CKIT=1883.,
  $
 CLASS 2 LRU NO. 4
 CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
  $L TI=.8,TIR=1.8,TD=.8,TDR=1.8,E=.001,TMD=.3,TMDR=.6,CUP=12250.,P=4.,
  WU=40.,WM=2.,WP=.1,CUBEU=.75,CUBEM=.02,CUBEP=.005,PP=40.,CMP=1360.,
  CPP=9.,GS=.85,GT=.15,CKIT=1225.,
  $
 CLASS 2 LRU NO. 5
 CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
  $L TI=1.,TIR=1.8,TD=1.,TDR=1.8,TMD=.1,TMDR=.4,WU=36.,IO=1,
  CUP=5000.,P=4.,PP=40.,CMP=1000.,CPP=6.,E=.0008,CKIT=500.,
  CI=0.,CPI=131500.,CRI=6000.,ETI=1.,GS=.85,GT=.15,
  $
 CLASS 3 LRU NO. 1
 CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
  $L E=.001,P=12.,PP=50.,TI=.5,TIR=1.6,TD=.5,TDR=1.6,TMD=.3,TMDR=.6,
  WU=30.,WM=1.5,WP=.08,CUBEU=1.,CUBEM=.05,CUBEP=.01,CKIT=2772.,
  TAYZ=1.,2*0.,7*1.,
  CPI=0.,CRI=0.,
  CUP=27716.,CMP=1610.,CPP=6.,ZI=0.,         GT=1.,
  $
```

7.1 LOGAM Sample Input Case (Continued)

```
SENSY ON FAIL RATE
FAILURE RATE IS  2- AND 3- TIMES BASELINE.
 $L SENSY=1.,2.,4.,81.,2.,3.,
 INHIB=1,IFLAG=1 $
END
FINIS
 $L NU=-4,IOPER=1, $

CLASS 3 LRU NO. 2
CASE I-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS
 $L NU=-1,IS=1,        ETII=1.,E=.0013,P=13.,PP=40.,TI=0.,WTKIT=30.,
 CUBEU=15.,CUBEM=.5,CUBEP=.05,WP=.5,CII=0.,CRII=7500.,CKIT=7526.,IO=2,
 CI=0.,CPI=0.,CRI=0.,ZI=0.,TMDR=3.4,CPP=11.,CPII=264000.,
 CCSP=0.,CCSPP=100000.,CONTCT=0.,CCSPR=1000.,EACSP=1.,ETI=1.,
 IO=3,
 NU=-3,
 CONTCT=5.,
 TD=1.,TMD=.75,CMP=2500.,TDR=3.5,TIR=3.5,WM=15.,CUP=75262.,WU=150.,GT=1.,
 $
```

7.2 Post Processor Sample Input

```
$TOE
T=1.,1.,0.,5.,1.,0.,0.,0.,0.,29804.,
1.,10.,0.,4.,1.,0.,0.,0.,0.,29804.,
1.,13.,0.,3.,1.,0.,0.,0.,0.,18718.,
1.,16.,0.,3.,0.,0.,1.,0.,0.,14038.5,
1.,8.,0.,2.,1.,0.,0.,0.,0.,21057.75,
1.,1.,0.,2.,0.,0.,0.,0.,1.,18718.,
1.,16.,0.,2.,0.,0.,1.,0.,0.,18718.,
1.,16.,0.,.5,0.,0.,0.,0.,1.,18080.,
1.,3.,0.,.5,1.,0.,0.,0.,0.,18080.,
1.,1.,1.,9.,1.,0.,0.,0.,0.,16759.,
1.,4.,1.,8.,1.,0.,0.,0.,0.,16765.75,
1.,4.,1.,8.,0.,0.,1.,0.,0.,16765.75,
1.,3.,1.,8.,0.,0.,0.,0.,1.,16759.,
1.,6.,1.,7.,1.,0.,0.,0.,0.,16759.,
1.,20.,1.,7.,0.,0.,1.,0.,0.,16759.,
1.,15.,1.,7.,0.,0.,0.,1.,0.,16759.,
1.,16.,1.,7.,0.,0.,0.,0.,1.,16759.,
1.,8.,1.,6.,1.,0.,0.,0.,0.,10469.,
1.,32.,1.,6.,0.,0.,1.,0.,0.,10469.,
1.,22.,1.,6.,0.,0.,0.,1.,0.,10469.,
1.,6.,1.,6.,0.,0.,0.,0.,1.,10469.,
1.,2.,1.,5.,0.,1.,0.,0.,0.,10469.,
1.,148.,1.,5.,0.,0.,1.,0.,0.,10469.,
1.,77.,1.,5.,0.,0.,0.,1.,0.,10469.,
1.,216.,1.,4.,0.,0.,1.,0.,0.,10469.,
1.,135.,1.,4.,0.,0.,0.,1.,0.,10469.,
1.,224.,1.,3.,0.,0.,1.,0.,0.,7500.,
1.,127.,1.,3.,0.,0.,0.,1.,0.,7500.,
2,1,.48,1769,247732,.295,8283,.295,7388.7,80052,
2,2,.112,17068,.112,17068,.036,.036,687,741,
2,3,.46,12011,0,0,0,0,0,0,
3,0,0,600,5,1.32,.0001,0,0,200,
3,0,0,400,4,1.32,.0001,0,0,62,
3,0,0,400,4,1.12,.0001,0,0,53,
3,0,0,324,138,1.50,.001,0,0,4,
4,16000,303360,5000,50000,0,0,0,0,0,
6,1,1000,50000,5000,10000,10000,0,0,3,
6,2,10000,50000,10000,15000,10000,4000,0,3,
7.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
8.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
$
```

SECTION 8

SAMPLE OUTPUTS

8.1 Individual LRU Case Output

```
                                   -29- (1)
               COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
          (2)   USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
                 MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                     ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

(3) UNIT - CLASS 3 LRU NO. 2                                              (4) ANALYSIS - THREE LRU CLASSES
(5) CASE 1-CONUS  REPAIR CL.1 AND CL.3 LRUS AT DEPOT-CL.2 LRUS AT DS      (6) DATE - JULY 1980

   PRESENT VALUE COST TOTAL
   EACH        CUM                                                  (10)                  (11)
   (7) 2775.  (8) 7352.  (9) THOUSANDS OF DOLLARS           AVAILABILITY= .999894  INHERENT= .999915
   PRIME      T.E.      TE SPACE MANPOWER  SUPPLY    ORDERING  STORAGE   S.ADMIN   SHIPPING  TOTAL
      (12) 0.    (13) 551.   (14) 0.   (15) 65.   (16) 2081.   (17) 1.   (18) 0.   (19) 73.   (20) 4.   (21) 2775.

   PROVISION          INITIAL BUY          REORDER BUY          CONSUMED             RESIDUAL
   UNIT  MODULE PART UNIT  MODULE PART UNIT  MODULE PART UNIT  MODULE PART UNIT  MODULE PART
   (22) 15.  (23) 1.  (24) 1.  (25) 56.  (26) 57.  (27) 58.  (28) 20.  (29) 50.  (30) 100.  (31) 0.  (32) 0.  (33) 5.  (34) 16.  (35) 1.  (36) 0.

   TEST EQUIPMENT AND REPAIR CHANNEL DATA
               DIRECT                                       GENERAL                                    DEPOT
      T.E.                         REPAIR         T.E.                           REPAIR       T.E.                                 REPAIR
(37)  EACH   (38)  CUM   (39) EACH   (40)  CUM   (41) EACH   (42)  CUM   (43) EACH   (44)  CUM   (45) EACH   (46)  CUM   (47) EACH   (48)  CUM
      0.0000       0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      .0063      .0063      .0211      .0211
             (49) 0.0000            (50) 0.0000          (51) .0051               (52) .0061              (53) .0272               (54) .0526

   ROUNDED-UP TOTALS FOR TYPE I TEST EQUIP., CHANNELS
          DIRECT                            GENERAL                                  DEPOT
   T.E.       TE MEN     REP MEN     T.E.      TE MEN     REP MEN     T.E.      TE MEN     REP MEN
 (55) 0.   (56) 0.    (57) 0.    (58) 0.    (59) 0.    (60) 0.    (61) 1.    (62) 0.    (63) 0.
(64) 0.0000                (65) 0.0000                       (66) .0070
   ROUNDED-UP TOTALS FOR TYPE II TEST EQUIP., CHANNELS
           DEPOT
   T.E.      TE MEN     REP MEN
    (67) 0.   (68) 0.    (69) 0.
   (70) 0.0000

   EXPECTED VALUE MANPOWER AT EQUIPMENT, DIRECT AND GENERAL

   PRESENT VALUE COST TOTAL
   EACH        CUM
   (71) 2775.  (72) 7352.     MANPOWER   (73) 65.     DELTA      (74) -0.     PV DELTA   (75) -0.
   INITIAL PROVISION QUANTITIES OF
                      UNITS                                            MODULES                                     PARTS
     EQPT.       DIRECT    GENERAL       DEPOT          DIRECT     GENERAL      DEPOT          DIRECT      GENERAL      DEPOT
   (76) 0.    (77) 4.    (78) 1.      (79) 0.       (80) 0.    (81) 0.    (82) 1.       (83) 0.     (84) 0.     (85) 1.
   COST OF INITIAL PROVISION                                                   RESIDUAL
            EQPT.      DIRECT      GENERAL       DEPOT        TOTAL
UNIT      (86) 0.   (87) 321.   (88) 333.    (89) 0.    (90) 1204.   (91) 1204.
MODULE              (92)   0.   (93)   0.    (94) 33.   (95)   33.   (96)   33.
PART                (97)   0.   (98)   0.    (99)  0.   (100)   0.   (101)   0.
```

8.2 Summarized LRU Case Output

```
                                        -52-(1)
                  COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
             (2)   USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
                    MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                       ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.
(3) UNIT - CLASS 3 LRU NO. 2                                   (4) ANLYSIS - ,THREE LRU CLASSES
  (5) TOTAL                                                    (6) DATE - JULY 1980

  PRESENT VALUE COST TOTAL
  EACH        CUM
   (7) 11141.  (8) 32402.   (9) THOUSANDS OF DOLLARS
  PRIME      T.E.       TE SPACE MANPOWER  SUPPLY    ORDERING  STORAGE   S.ADMIN   SHIPPING  TOTAL
     (12) 0.  (13) 3630.   (14) 0.   (15) 212.  (16) 6957.   (17) 4.   (18) 0.  (19) 258.  (20) 19.  (21) 11141.

  PROVISION            INITAL BUY                                CONSUMED              RESIDUAL
  UNIT  MODULE PART UNIT  MODULE PART                        UNIT  MODULE PART UNIT  MODULE PART
  (22) 45. (23) 5. (24) 3. (25) 226. (26) 231. (27) 234.    (31) 0. (32) 0. (33) 25. (34) 45. (35) 5. (36) 0.

  TEST EQUIPMENT AND REPAIR CHANNEL DATA
               DIRECT                                   GENERAL                                     DEPOT
      T.E.                           REPAIR     T.E.                         REPAIR     T.E.                          REPAIR
      EACH          CUM       EACH      CUM     EACH        CUM       EACH      CUM     EACH        CUM       EACH       CUM
  (37) 0.0000 (38) 0.0000 (39) 0.0000 (40) 0.0000 (41) .0097 (42) .0097 (43) .0273 (44) .0273 (45) .0181 (46) .0181 (47) .0672 (48) .0672
             (49) 0.0000          (50) 0.0000         (51) .0417          (52) .0736          (53) .0926          (54) .1791

  ROUNDED-UP TOTALS FOR TYPE I TEST EQUIP., CHANNELS
               DIRECT                         GENERAL                          DEPOT
  T.E.      TE MEN     REP MEN    T.E.      TE MEN     REP MEN    T.E.       TE MEN     REP MEN
  (55) 0.   (56) 0.    (57) 0.    (58) 1.   (59) 0.    (60) 0.    (61) 2.    (62) 0.    (63) 0.
(64) 0.0000              (65) .0107                      (66) .0199
  ROUNDED-UP TOTALS FOR TYPE II TEST EQUIP., CHANNELS
                 DEPOT
  T.E.      TE MEN      REP MEN
     (67) 0.    (68) 0.    (69) 0.
   (70) 0.0000

  EXPECTED VALUE MANPOWER AT EQUIPMENT, DIRECT AND GENERAL

  PRESENT VALUE COST TOTAL
  EACH        CUM
  (71) 11141. (72) 32402.   MANPOWER  (73) 212.   DELTA   (74) -0.   PV DELTA  (75) -0.
  INITIAL PROVISION QUANTITIES OF
                  UNITS                                          MODULES                                       PARTS
     EQPT.      DIRECT      GENERAL      DEPOT        DIRECT     GENERAL     DEPOT        DIRECT     GENERAL      DEPOT
  (76) 0.   (77) 22.    (78) 23.     (79) 0.      (80) 0.    (81) 3.     (82) 2.      (83) 0.    (84) 0.      (85) 3.
  COST OF INITIAL PROVISION                                              RESIDUAL
          EQPT.      DIRECT       GENERAL      DEPOT       TOTAL
UNIT    (86) 0.   (87) 1656.   (88) 1731.   (89) 0.    (90) 3387.   (91) 3387.
MODULE            (92) 0.      (93) 98.     (94) 65.   (95) 163.    (96) 163.
PART              (97) 0.      (98) 0.      (99) 1.    (100) 1.     (101) 0.
```

8.3.1 Individual LRU Supplementary Output

```
                                   -10-
            COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
             USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
              MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                 ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

UNIT - CLASS 2 LRU NO. 1                                                ANALYSIS - THREE LRU CLASSES
CASE I-USAREUR REPAIR CL.1 LRUS AT DEPOT-CL.2 LRUS AT DS-CL.3 LRUS AT GS   DATE - JULY 1980

               SUPPLEMENTARY INFORMATION REGARDING POLICIES GC, GI, GJ GK/

TEST EQUIPMENT AND REPAIR CHANNEL DATA
FOR EQUIPMENT LOCATED FACILITIES
      T.E.                                  REPAIR
      EACH          CUM          EACH          CUM
(1)  .0001    (2) .0001    (3) .0001    (4) .0001
              (5) .0002                 (6) .0002

ROUNDED UP TOTALS FOR TYPE I TEST EQUIPMENT CHANNELS AT EQUIPMENT LOCATION

     T.E.        TE MEN      REP MEN
(7)  141.    (8) 0.      (9) 0.
(10) .0001
     INITIAL PROVISION QUANTITIES OF MODULES AT EQUIPMENT -   (11) 141.
     COST OF INITIAL PROVISION MODULES AT THE EQUIPMENT -     (12) 4399.
AT CHECK    5  0.          0.          0.          0.          0.          -.13642421E-11  -.36379788E-11
```

8.3.2 Summarized LRU Supplementary Output

```
                                        -53-
                      COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
                      USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
                       MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                          ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.
UNIT - CLASS 3 LRU NO. 2                                           ANLYSIS - ,THREE LRU CLASSES
 TOTAL                                                              DATE - JULY 1980


                   SUPPLEMENTARY INFORMATION REGARDING POLICIES GC, GI, GJ GK/


TEST EQUIPMENT AND REPAIR CHANNEL DATA
FOR EQUIPMENT LOCATED FACILITIES
      T.E.                                   REPAIR
  (1) EACH     (2)   CUM   (3)  EACH    (4)    CUM
    0.0000         0.0000     0.0000        0.0000
               (5) .0002            (6) .0002


ROUNDED UP TOTALS FOR TYPE I TEST EQUIPMENT CHANNELS AT EQUIPMENT LOCATION

T.E.         TE MEN     REP MEN
(7) 0.     (8) 0.     (9) 0.
(10) 0.0000
INITIAL PROVISION QUANTITIES OF MODULES AT EQUIPMENT -        (11) 0.
COST OF INITIAL PROVISION MODULES AT THE EQUIPMENT -          (12) 0.
```

8.4.1 Case Total Output

```
                                          -69-
                COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
                USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
                 MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                   ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

ANALYSIS - THREE LRU CLASSES                                                        DATE - JULY 1979
                                                CASE TOTAL
          (1)
COST TOTALS, COST IN THOUSANDS OF DOLLARS                                           RECURRING COSTS
     INSTALLED EQUIPMENT                 0.(2)
     TEST EQUIPMENT                   1531.(3)                                  T.E. MAINTENANCE        374. (4)
     TEST EQUIPMENT SPACE                0.(5)                            DEPOT SPACE/UTILITIES           0. (6)
     MAINTENANCE MANPOWER              695.(7)          FIELD   (8) 223.  DEPOT   (9) 444. TOTAL        667.(10)
                                              TRAINING FIELD   (11)  13.  DEPOT  (12)   0. TOTAL         13.(13)
     SUPPLY MATERIAL                  8496.(14)                                        SUPPLIES        2841.(15)
     REORDERING                         21.(16)                                      REORDERING          21.(17)
     MATERIAL STORAGE                    0.(18)                                MATERIAL STORAGE           0.(19)
     SUPPLY ADMINISTRATION             585.(20)                            INVENTORY MANAGEMENT         462.(21)
     SHIPPING AND HANDLING              11.(22)                                        SHIPPING          11.(23)

     GRAND TOTAL COST                11339. (24)                                TOTAL RECURRING        4330. (25)


     PRESENT VALUE                                                     COST OF INITIAL PROVISION
     DEVELOPMENT                         0. (26)                                          UNITS        5243. (27)
     ACQUISITION                      6935. (28)                                        MODULES         409. (29)
     OPERATION AND MAINTENANCE        4404. (30)                                          PARTS           2. (31)
     END LIFE SALVAGE                    0. (32)                                TOTAL PROVISION        5655. (33)

     GRAND TOTAL                     11339. (34)

EXPECTED VALUE MANPOWER AT EQUIPMENT DIRECT AND GENERAL

     MAINTENANCE MANPOWER              695. (35)
     GRAND TOTAL COST                11339. (36)
     PRESENT VALUE
     OPERATION AND MAINTENANCE        4404. (37)
     GRAND TOTAL                     11339. (38)      DELTA        -0. (39)      PV DELTA        -0. (40)
```

8.4.2 Grand Total Output

```
                                          -71-
            COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
             USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
              MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

ANALYSIS - THREE LRU CLASSES               GRAND TOTAL                         DATE - JULY 198?

COST TOTALS, COST IN THOUSANDS OF DOLLARS (1)                                  RECURRING COSTS
     INSTALLED EQUIPMENT               0. (2)
     TEST EQUIPMENT                 7113. (3)                           T.E. MAINTENANCE      727. (4)
     TEST EQUIPMENT SPACE              0. (5)                    DEPOT SPACE/UTILITIES          0. (6)
     MAINTENANCE MANPOWER           2265. (7)          FIELD (8) 1022.  DEPOT (9) 1136. TOTAL   2158. (10)
                                             TRAINING FIELD (11)   60.  DEPOT (12)   0. TOTAL     60. (13)
     SUPPLY MATERIAL               31078. (14)                                  SUPPLIES    10733. (15)
     REORDERING                       96. (16)                                REORDERING       96. (17)
     MATERIAL STORAGE                  0. (18)                          MATERIAL STORAGE        0. (19)
     SUPPLY ADMINISTRATION          2010. (20)                      INVENTORY MANAGEMENT     1605. (21)
     SHIPPING AND HANDLING            53. (22)                                  SHIPPING       53. (23)

     GRAND TOTAL COST              42620. (24)                           TOTAL RECURRING    15432. (25)


     PRESENT VALUE                                                  COST OF INITIAL PROVISION
     DEVELOPMENT                    3619. (26)                                     UNITS    13186. (27)
     ACQUISITION                   23522. (28)                                   MODULES     7149. (29)
     OPERATION AND MAINTENANCE     15479. (30)                                     PARTS       10. (31)
     END LIFE SALVAGE                  0. (32)                           TOTAL PROVISION    20345. (33)

     GRAND TOTAL                   42620. (34)

EXPECTED VALUE MANPOWER AT EQUIPMENT DIRECT AND GENERAL

     MAINTENANCE MANPOWER           2265. (35)
     GRAND TOTAL COST              42620. (36)
     PRESENT VALUE
     OPERATION AND MAINTENANCE     15479. (37)
     GRAND TOTAL                   42620. (38)     DELTA       -0. (39)     PV DELTA      -0. (40)
```

8.5 Maintenance Output

```
                                     $$$$$IMPORTANT$$$$$
THE FOLLOWING SEVEN CHECK NUMBERS MUST BE ZERO FOR THE PAM COST EQS. FROM LOGAM TO BE CORRECT.
CHECK EQS SHOULD BE ZERO IF PAM EQS ARE OK
 -.22737368E-12  0.            0.               -.14551915E-10  0.            -.68212103E-12  -.72759576E-11
      (1)       (2)           (3)                    (4)        (5)                (6)             (7)

                       $$$$$$$$$THE TOTAL LOGAM COST MUNUS PAM COST EQUALS     -.87311491E-10   $$$$$$$$$
                                                                                     (8)

INDIVIDUAL COST CATEGORIES FROM LOGAM ADDED TO PAM BREAK-OUT
(9) 0.          (10) 0.         (11) 0.         (12) 0.          (13) .56546785E+04 (14).21732765E+03 (15).23000000E+03 (16).47150490E+03
(17) .88333832E+02 (18).41419062E+01 (19).29125360E+04  0.(20)   (21) .29573339E+02(22).71178000E+01(23).69661225E+03  0.(24)
(25) CAVZ=  .997155   .999586   .996491   .999336
(26) CAVZI= .997330   .999763   .998679   .999546
                               MAN-E          EQUIP          DIRECT          GENERAL          DEPOT
HRS PER YR ALL MAINT LOC
   TEST EQUIPMENT            76.175 (27)     0.000 (28)     0.000 (29)     537.777 (30)     609.076 (31)
   REPAIR                   634.791 (32)     0.000 (33)     0.000 (34)     598.133 (35)    1220.798 (36)
NO OF MEN PER ANY TIME
UNIT ALL MAINT LOC
   TEST EQUIPMENT              .030 (37)     0.000 (38)     0.000 (39)        .515 (40)        .584 (41)
   REPAIR                      .253 (42)     0.000 (43)     0.000 (44)        .573 (45)       1.170 (46)
HRS PER YR PER MAINT LOC
   TEST EQUIPMENT             1.904 (47)     0.000 (48)     0.000 (49)     134.444 (50)     609.076 (51)
   REPAIR                    15.870 (52)     0.000 (53)     0.000 (54)     149.533 (55)    1220.798 (56)
NO OF MEN PER ANY TIME
UNIT PER MAINT LOC
   TEST EQUIPMENT              .001 (57)     0.000 (58)     0.000 (59)        .129 (60)        .584 (61)
   REPAIR                      .006 (62)     0.000 (63)     0.000 (64)        .143 (65)       1.170 (66)
```

8.6.1 System Maintenance Support Cost Output

-72-

COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

SYSTEM MAINTENANCE SUPPORT COSTS

			COST	PERCENTAGE	
1.000	RESEARCH AND DEVELOPMENT				
1.020	DEVELOPMENT ENGINEERING	(1)	3619.00	100.00	(2)
TOTAL		(3)	3619.00	100.00	(4)
2.000	INVESTMENT COST				
2.010	NON-RECURRING INVESTMENT	(5)	0.00	0.00	(6)
2.050	DATA	(7)	0.00	0.00	(8)
2.070	TRAINING SERVICES AND EQUIPMENT	(9)	0.00	0.00	(10)
2.090	INITIAL SPARES AND REPAIR PARTS	(11)	20345.25	88.01	(12)
2.11	OTHER	(13)	2772.00	11.99	(14)
TOTAL		(15)	23117.25	100.00	(16)
3.000	OPERATING AND SUPPORT COST				
3.010	MILITARY PERSONNEL				
3.011	CREW PAY AND ALLOWANCES	(17)	0.00	0.00	(18)
3.012	MAINTENANCE PAY AND ALLOWANCES	(19)	1019.31	6.42	(20)
3.013	INDIRECT PAY AND ALLOWANCES	(21)	0.00	0.00	(22)
3.014	PERMANENT CHANGE OF STATION	(23)	0.00	0.00	(24)
3.020	CONSUMPTION				
3.021	REPLENISHMENT SPARES	(25)	533.00	3.36	(26)
3.022	PETROLEUM, OIL AND LUBRICANTS	(27)	0.00	0.00	(28)
3.023	UNIT TRAINING AMMUNITION AND MISSILE	(29)	0.00	0.00	(30)
3.030	DEPOT MAINTENANCE				
3.031	LABOR	(31)	1207.35	7.60	(32)
3.032	MATERIEL	(33)	245.77	1.55	(34)
3.033	TRANSPORTATION	(35)	4.18	.03	(36)
3.040	MODIFICATIONS MATERIAL	(37)	10603.77	66.76	(38)
3.050	OTHER DIRECT SUPPORT OPERATIONS				
3.051	MAINTENANCE, CIVILIAN LABOR	(39)	0.00	0.00	(40)
3.052	OTHER DIRECT		2106.24	13.26	(42)
3.060	INDIRECT SUPPORT OPERATIONS				
3.061	PERSONNEL REPLACEMENT	(43)	115.43	.73	(44)
3.062	TRANSIENTS, PATIENTS AND PRISONERS	(45)	0.00	0.00	(46)
3.063	QUARTERS, MAINTENANCE AND UTILITIES	(47)	0.00	0.00	(48)
3.064	MEDICAL SUPPORT	(49)	0.00	0.00	(50)
3.065	OTHER INDIRECT	(51)	48.61	.31	(52)
TOTAL		(53)	15883.67	100.00	(54)
GRAND TOTAL		(55)	42619.92		

8.6.2 System Operation and Support Cost Output

-72-

COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

SYSTEM OPERATIONS AND SUPPORT COSTS

			COST	PERCENTAGE	
1.000	RESEARCH AND DEVELOPMENT				
1.010	DEVELOPMENT ENGINEERING	(1)	3619.00	100.00	(2)
TOTAL		(3)	3619.00	100.00	(4)
2.000	INVESTMENT COST				
2.010	NON-RECURRING INVESTMENT	(5)	0.00	0.00	(6)
2.050	DATA	(7)	0.00	0.00	(8)
2.060	TRAINING SERVICES AND EQUIPMENT	(9)	0.00	0.00	(10)
2.030	INITIAL SPARES AND REPAIR PARTS	(11)	20345.25	88.01	(12)
2.11	OTHER	(13)	2772.00	11.99	(14)
TOTAL		(15)	23117.25	100.00	(16)
3.000	OPERATING AND SUPPORT COST				
3.010	MILITARY PERSONNEL				
3.011	CREW PAY AND ALLOWANCES	(17)	67520.71	5.35	(18)
3.012	MAINTENANCE PAY AND ALLOWANCES	(19)	1019.31	.08	(20)
3.013	INDIRECT PAY AND ALLOWANCES	(21)	14831.13	1.18	(22)
3.014	PERMANENT CHANGE OF STATION	(23)	408.33	.03	(24)
3.020	CONSUMPTION				
3.021	REPLENISHMENT SPARES	(25)	533.00	.04	(26)
3.022	PETROLEUM, OIL AND LUBRICANTS	(27)	1.20	.00	(28)
3.023	UNIT TRAINING AMMUNITION AND MISSILE	(29)	13.19	.00	(30)
3.030	DEPOT MAINTENANCE				
3.031	LABOR	(31)	1207.35	.10	(32)
3.032	MATERIEL	(33)	245.77	.02	(34)
3.033	TRANSPORTATION	(35)	4.18	.00	(36)
3.040	MODIFICATIONS MATERIAL	(37)	10603.77	.84	(38)
3.050	OTHER DIRECT SUPPORT OPERATIONS				
3.051	MAINTENANCE, CIVILIAN LABOR	(39)	0.00	0.00	(40)
3.052	OTHER DIRECT	(41)	2108.38	.17	(42)
3.060	INDIRECT SUPPORT OPERATIONS				
3.061	PERSONNEL REPLACEMENT	(43)	571.79	.05	(44)
3.062	TRANSIENTS, PATIENTS AND PRISONERS	(45)	1161343.53	92.12	(46)
3.063	QUARTERS, MAINTENANCE AND UTILITIES	(47)	14.28	.00	(48)
3.064	MEDICAL SUPPORT	(49)	259.07	.02	(50)
3.065	OTHER INDIRECT	(51)	48.75	.00	(52)
TOTAL		(53)	1261234.33	100.00	(54)
GRAND TOTAL		(55)	1287970.58		

8.7.1 Sensitivity Output (Inhibited print)

```
                                          -56-
              COMPARISON OF FIELD VERSUS DEPOT SUPPORT FOR SELECTED MICOM MISSILE LRUS
              USING LIFE CYCLE COST OF OWNERSHIP AND OPERATIONAL AVAILABILITY AS THE
               MEASURES OF EFFECTIVENESS. THE SYSTEM AVAILABILITY PRODUCT CONSIDERS
                  ONLY THOSE LRUS WHICH OPERATE TOGETHER AS A FUNCTIONAL GROUP.

  UNIT - SENSY ON FAIL RATE (1)                                          ANALYSIS - THREE LRU CLASSES
  FAILURE RATE IS  2- AND 3- TIMES BASELINE. (2)                          DATE - JULY 1980

                   (3)         (4)    (5)        (6)
SENSITIVITY PASS    1  LRU    1  VAR( 81)=  .2000000E-03  VAR(
AT CHECK   23  0.                0.              0.            0.              0.              -.11369694E-12  0.
SENSITIVITY PASS    1  LRU    2  VAR( 81)=  .2000000E-03  VAR(
AT CHECK   24  0.                0.              0.            -.90949470E-12  0.              -.11368694E-12  0.
SENSITIVITY PASS    1  LRU    3  VAR( 81)=  .1000000E-02  VAR(
AT CHECK   25  0.                0.              0.            -.90949470E-12  0.              -.45474735E-12  0.
SENSITIVITY PASS    1  LRU    4  VAR( 81)=  .1600000E-02  VAR(
AT CHECK   26  0.                0.              0.            0.              0.              -.45474735E-12  0.
SENSITIVITY PASS    1  LRU    5  VAR( 81)=  .4200000E-02  VAR(
AT CHECK   27  0.                0.              0.            -.29103830E-10  0.              -.15916157E-11  -.72759576E-11
SENSITIVITY PASS    1  LRU    6  VAR( 81)=  .3400000E-02  VAR(
AT CHECK   28  0.                0.              0.            0.              0.              -.40927262E-11  -.36379788E-11
SENSITIVITY PASS    1  LRU    7  VAR( 81)=  .2200000E-02  VAR(
AT CHECK   29  0.                0.              0.            -.36379788E-11  0.              -.15916157E-11  -.36379788E-11
SENSITIVITY PASS    1  LRU    8  VAR( 81)=  .2000000E-02  VAR(
AT CHECK   30  0.                0.              0.            -.18189894E-11  0.              -.14779299E-11  -.18189894E-11
SENSITIVITY PASS    1  LRU    9  VAR( 81)=  .1600000E-02  VAR(
AT CHECK   31  -.90949470E-12    0.              0.            0.              0.              -.15916157E-11  -.90949470E-12
SENSITIVITY PASS    1  LRU   10  VAR( 81)=  .2000000E-02  VAR(
AT CHECK   32  0.                0.              0.            0.              0.              -.10231815E-11  -.36379788E-11
SENSITIVITY PASS    1  LRU   11  VAR( 81)=  .2600000E-02  VAR(
AT CHECK   33  -.45474735E-12    0.              0.            -.14551915E-10  0.              -.25011104E-11  -.14551915E-10
```

8

DTIC